



## 用例 NetApp Solutions

NetApp  
September 10, 2024

# 目录

用例 .....	1
负责的AI和机密推理—NetApp AI与Protopia Image Transformation .....	1
使用 NetApp AI 进行情感分析 .....	24
Azure 中的分布式培训—点击率预测 .....	41
TR-4896： Azure 中的分布式培训：通道检测—解决方案设计 .....	64
TR-4841： 采用数据缓存的混合云 AI 操作系统 .....	91
TR-4732： 大数据分析数据到人工智能 .....	112
边缘人工智能推理—采用联想 ThinkSystem 的 NetApp —解决方案设计 .....	114
WP-7328： 使用 NVIDIA JarVis 的 NetApp 对话 AI .....	135
TR-4858： 《 NetApp Orchestration 解决方案与 Run： AI 》 .....	154
TR-4799-design： 适用于自主驱动工作负载的NetApp ONTAP AI参考架构 .....	172
TR-4811： 《适用于医疗保健的NetApp ONTAP AI参考架构：诊断映像—解决方案 设计》 .....	172
TR-4807： 适用于金融服务工作负载的NetApp ONTAP AI参考架构—解决方案 设计 .....	172
生成性AI和NetApp价值 .....	173
TR-4785： 使用NetApp E系列和BeeGFS进行AI部署 .....	181
NVA-1150-设计： 《采用NetApp E系列系统的Quantum StorNext》设计指南 .....	181
NVA-1150-Deploy： 《采用NetApp E系列系统的Quantum StorNext部署指南》 .....	182

# 用例

## 负责的AI和机密推理—NetApp AI与Protopia Image Transformation

### TR-4928：《责任AI和机密推理—NetApp AI与Protopia映像和数据转型》

Sathish Thyagarajan、Michael Oglesby、NetApp ByG Hoon Ahn、Jennifer Cwagenerberg、Protopia

随着图像捕获和图像处理的出现、视觉解释已成为通信不可或缺的一部分。数字图像处理中的人工智能(AI)带来了新的业务机会、例如癌症和其他疾病的医学领域、用于研究环境危害的地理空间视觉分析、模式识别、用于打击犯罪的视频处理等。但是、这一机会也伴随着非凡的责任。

企业将越来越多的决策交由AI处理、他们就越能接受与数据隐私和安全以及法律、道德和法规问题相关的风险。通过负责的AI、企业和政府组织可以建立信任和监管、这对于大型企业的大规模AI至关重要。本文档介绍了在三种不同情形下由NetApp验证的AI推理解决方案、它使用NetApp数据管理技术与Protopia数据混淆软件将敏感数据私营化并降低风险和道德问题。

消费者和业务实体每天都使用各种数字设备生成数百万张图像。因此、数据和计算工作负载的大规模激增使企业转向云计算平台以实现规模和效率。同时、在传输到公有云时、会出现有关图像数据中所含敏感信息的隐私问题。缺乏安全和隐私保证已成为部署图像处理AI系统的主要障碍。

此外、还有 **"擦除权"** 根据《一般数据保护条例》、个人有权要求组织擦除其所有个人数据。此外、还提供了 **"隐私法案"**、建立公平信息实践准则。根据《一般数据保护条例》、照片等数字图像可以构成个人数据、该条例规定了数据的收集、处理和擦除方式。如果不遵守《一般数据保护条例》、可能会因违反合规性而被处以巨额罚款、从而对组织造成严重损害。隐私原则是实施负责AI的基石、它可以确保机器学习(ML)和深度学习(DL)模型预测的公平性、并降低与违反隐私或法规相关的风险。

本文档介绍了在三种不同场景下经过验证的设计解决方案、其中包括与保护隐私和部署负责的AI解决方案相关的图像混淆和不混淆：

- \*场景1.\*在Jupyter笔记本电脑中按需推理。
- 场景2. Kubernetes上的批处理推理。
- 场景3. NVIDIA Triton推理服务器。

对于此解决方案、我们将使用面部检测数据集和基准测试(FDDB)、这是一个人脸区域数据集、用于研究无约束面部检测问题、并结合使用PyTorch机器学习框架来实施FaceBoxes。此数据集包含一组具有各种分辨率的2845图像中5171个面的标注。此外、本技术报告还介绍了解决方案的一些领域以及在适用此解决方案的情况下从NetApp客户和现场工程师收集的相关用例。

#### 目标受众

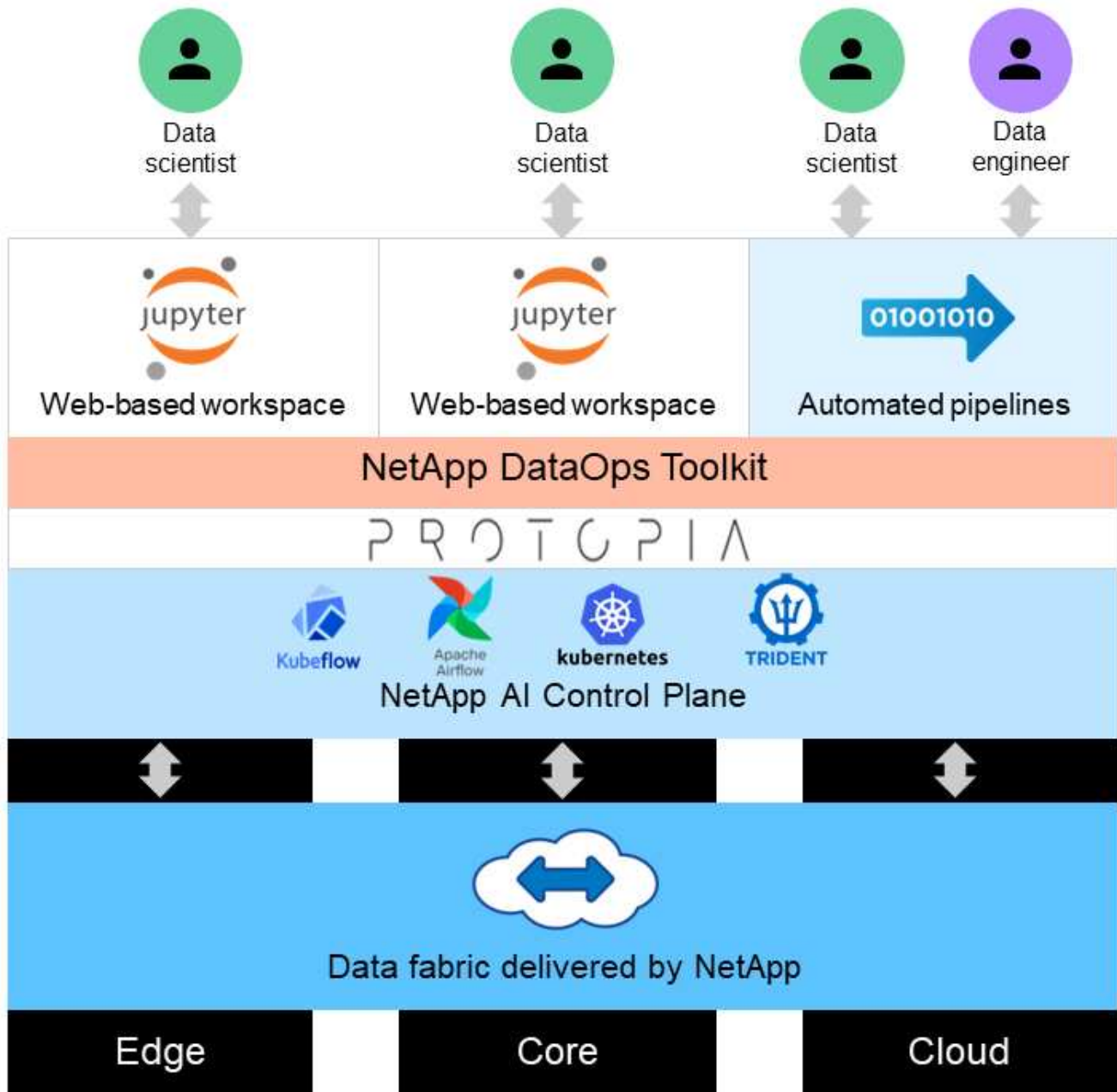
本技术报告面向以下受众：

- 业务主管和企业架构师、他们希望设计和部署负责的AI、并解决与公有空间中面部图像处理相关的数据保护和隐私问题。
- 数据科学家、数据工程师、AI/机器学习(ML)研究人员以及旨在保护和维护隐私的AI/ML系统开发人员。

- 为符合GDPR、CCPA或国防部(DoD)隐私法案等法规标准的AI/ML模型和应用程序设计数据混淆解决方案的企业架构师。
- 数据科学家和AI工程师正在寻找高效的方法来部署深度学习(DL)和AI/ML/DL推理模型、以保护敏感信息。
- 边缘设备管理器和边缘服务器管理员，负责部署和管理边缘推理模型。

## 解决方案架构

此解决方案 可通过利用GPU的处理能力以及传统CPU来处理大型数据集中的实时和批量推理AI工作负载。此验证证明了ML的隐私保护推理以及寻求负责AI部署的组织所需的最佳数据管理。此解决方案 提供的架构适合单节点或多节点Kubernetes平台、用于与核心内部部署的NetApp ONTAP AI、NetApp DataOps工具包以及使用Jupyter实验室和CLI界面的Protopia混淆软件互连的边缘和云计算。下图显示了由NetApp提供支持的Data Fabric与DataOps Toolkit和Protopia的逻辑架构概述。



Protopia混淆软件可在NetApp DataOps工具包上无缝运行、并在离开存储服务器之前转换数据。

## 解决方案区域

数字图像处理具有许多优势、可以使许多组织充分利用与可视化表示相关的数据。NetApp和Protopia解决方案 提供了一种独特的AI推理设计、用于在整个ML/DL生命周期内保护AI/ML数据并将其私有化。它可以帮助客户保留敏感数据的所有权、通过减轻与隐私相关的顾虑来使用公有 或混合云部署模式以实现规模和效率、并在边缘部署AI推理。

### 环境智能

行业可以通过多种方式在环境危害领域利用地理空间分析。政府和公有 Works部门可以获得有关公有 运行状况和天气状况的切实可行的见解、以便在发生大流行病或野火等自然灾害时更好地为公有 提供建议。例如、您可以在机场或医院等公有 空间识别出COVID-阳性患者、而不会影响受影响患者的隐私、并提醒相关主管部门和附近的公有 采取必要的安全措施。

### 边缘设备可穿戴设备

在军队和战场上、您可以使用边缘的人工智能推理作为可穿戴设备来跟踪军人的健康状况、监控司机行为、并就近军车的安全和相关风险向当局发出警报、同时保护和保护军人的隐私。随着战场物联网(Internet of Battlefield Things、IoBT)和军事物联网(Internet of militar Things、IoMT)的发展、军队的未来将采用高科技设备、通过快速边缘计算帮助军人识别敌人并提高作战性能。保护和保留从无人机和可穿戴设备等边缘设备收集的视觉数据对于防止黑客和敌人入侵至关重要。

### 非作战人员疏散操作

非作战人员后送行动(Noncombatant evelevation operations、NEO)由国防部执行、旨在帮助将生命面临危险的美国公民和公民、国防部文职人员以及指定人员(东道国(HN)和第三国公民(TCN))疏散到适当的安全天堂。所实施的管理控制主要采用手动疏散人员筛查流程。但是、使用高度自动化的AI/ML工具以及AI/ML视频模糊技术、可能会提高疏散人员识别、疏散人员跟踪和威胁筛选的准确性、安全性和速度。

### 医疗保健和生物医学研究

图像处理用于通过计算机断层成像(CT)或磁场共振成像(MRI)获得的3D图像诊断用于手术计划的病理学。HIPAA隐私规则规定了组织必须如何收集、处理和擦除所有个人信息和照片等数字图像的数据。要使数据符合HIPAA《安全港》规定的可共享性、必须删除全面照片图像和任何类似图像。用于从结构化的CT/MR图像中模糊个人面部特征的除名或头骨 - 剥离算法等自动化技术已成为生物医学研究机构数据共享过程的重要组成部分。

### 人工智能/机器学习分析的云迁移

传统上、企业客户已经在内部培训和部署了AI/ML模式。出于规模经济和效率方面的考虑、这些客户正在进行扩展、将AI/ML功能迁移到公有 、混合云或多云云部署中。但是、它们受数据可以公开到其他基础架构的约束。NetApp解决方案可解决所需的各种网络安全威胁 **"数据保护"** 和安全性评估、结合使用Protopia数据转换、最大程度地降低将图像处理AI/ML工作负载迁移到云的相关风险。

有关其他行业的边缘计算和AI推理的其他用例、请参见 ["TR-4886人工智能在边缘进行推理"](#) 以及NetApp AI博客、**"智能与隐私"**。

## 技术概述

本节概述了完成此解决方案 所需的各种技术组件。

## Protopia

Protopia AI提供了一种不引人注目的纯软件解决方案、用于在当今市场中进行保密推理。通过最大限度地减少敏感信息的暴露、Protopia解决方案 可为推理服务提供无与伦比的保护。AI只会馈送数据记录中对执行手头任务真正至关重要的信息、而不会再提供更多信息。大多数推理任务不会使用每个数据记录中的所有信息。无论您的AI是使用图像、语音、视频还是结构化表格数据、Protopia都能提供推理服务所需的功能。获得专利的核心技术利用数学精心设计的噪声来随机转换数据、并收集给定ML服务不需要的信息。此解决方案 不会屏蔽数据、而是会使用随机筛选的噪声来更改数据表示形式。

Protopia解决方案 将更改表示形式的问题表述为基于梯度的最大干扰方法、该方法仍会在输入功能空间中保留与模型功能相关的信息。此发现过程将在训练ML模型结束时作为微调过程运行。在传递自动生成一组概率分布之后、低开销数据转换会将这些分布中的噪声样本应用于数据、并在将其传递到模型进行推理之前使其发生混淆。

## NetApp ONTAP AI

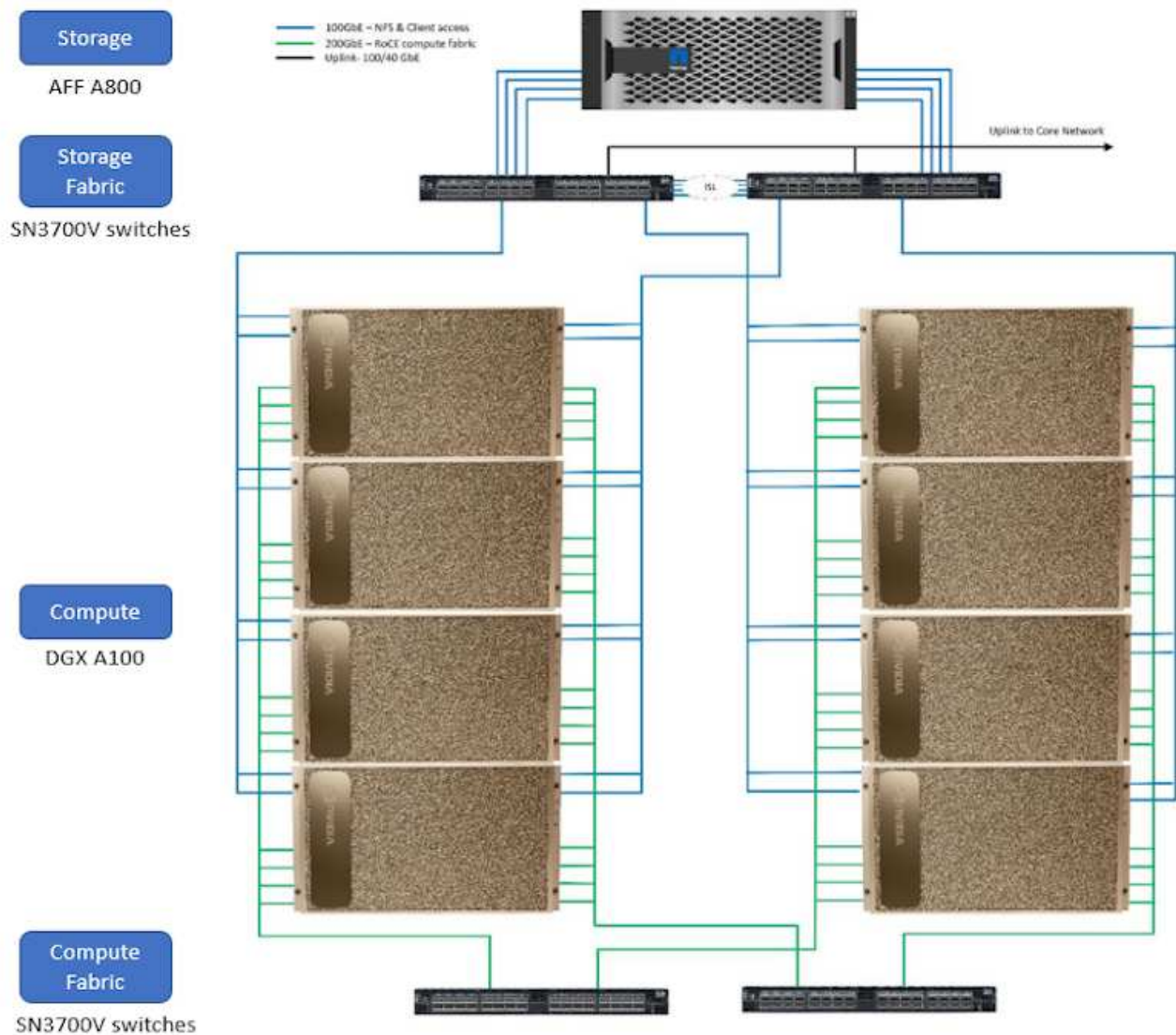
NetApp ONTAP AI参考架构由DGX A100系统和NetApp云互联存储系统提供支持、由NetApp和NVIDIA开发并验证。它为 IT 组织提供了一个架构，可提供以下优势：

- 消除设计复杂性
- 支持独立扩展计算和存储
- 支持客户从小规模入手，无缝扩展
- 为各种性能和成本点提供了一系列存储选项

ONTAP AI将DGX A100系统和NetApp AFF A800存储系统与一流的网络紧密集成在一起。ONTAP AI可消除设计复杂性和猜测性工作、从而简化AI部署。客户可以从小规模入手、实现无中断增长、同时智能管理从边缘到核心再到云再到云的数据。

下图显示了采用DGX A100系统的ONTAP AI解决方案系列中的多种变体。最多可使用八个DGX A100系统验证AFF A800系统性能。通过向ONTAP 集群添加存储控制器对、该架构可以扩展到多个机架、以支持多个DGX A100系统和数PB的存储容量、并实现线性性能。这种方法可以灵活地根据所使用的DL型号的大小以及所需的性能指标独立更改计算与存储的比率。





有关追加信息 关于ONTAP AI的信息、请参见 ["NVA-1153：采用NVIDIA DGX A100系统和Mellanox频谱以太网交换机的NetApp ONTAP AI。"](#)

## NetApp ONTAP

ONTAP 9.11是NetApp推出的最新一代存储管理软件、可帮助企业打造现代化的基础架构并过渡到云就绪数据中心。借助行业领先的数据管理功能，无论数据位于何处，ONTAP 都可以通过一组工具来管理和保护数据。您还可以将数据自由移动到需要的任何位置：边缘，核心或云。ONTAP 9.11包含许多功能、可简化数据管理、加快和保护关键数据、并在混合云架构中实现下一代基础架构功能。

## NetApp DataOps 工具包

NetApp DataOps工具包是一个Python库、可使开发人员、数据科学家、开发运营工程师和数据工程师轻松执行各种数据管理任务、例如近乎即时地配置新的数据卷或JupyterLab工作空间、近乎即时地克隆数据卷或JupyterLab工作空间、并近乎即时地为数据卷或JupyterLab工作空间创建快照、以实现可追溯性或基线化。此Python库可以用作命令行实用程序或函数库、您可以将其导入到任何Python程序或Jupyter笔记本电脑中。

## NVIDIA Triton 推理服务器

NVIDIA Triton推理服务器是一款开源推理服务软件、可帮助实现模型部署和执行标准化、从而在生产环境中提供快速且可扩展的AI。Triton推理服务器支持团队从任何基于GPU或CPU的基础架构上的任何框架部署、运行和扩展经过培训的AI模型、从而简化了AI推理过程。Triton推理服务器支持所有主要框架、例如TensorFlow、NVIDIA TensorRT、PyTorch、MXNet、OpenVINO等。Triton与Kubernetes集成、用于协调和扩展、您可以在所有主要公有云AI和Kubernetes平台中使用。它还与许多MLOps软件解决方案集成在一起。

## PyTorch

"PyTorch" 是开源ML框架。它是一个经过优化的用于深度学习的tensor库、使用GPU和CPU。PyTorch软件包包含多维感应器的数据结构、可提供许多实用程序来高效地序列化其他有用实用程序中的感应器。它还具有一个CUDA对应项、可用于在具有计算功能的NVIDIA GPU上运行Tensor计算。在此验证中、我们使用OpenCV-Python (CV2)库来验证我们的型号、同时利用Python最直观的计算机视觉概念。

### 简化数据管理

数据管理对于企业IT运营和数据科学家至关重要、这样才能将适当的资源用于AI应用程序和训练AI/ML数据集。以下有关NetApp技术的追加信息 不在此验证范围内、但可能与您的部署相关。

ONTAP 数据管理软件包括以下功能、可简化操作并降低总运营成本：

- 实时数据缩减和扩展的重复数据删除。数据缩减可减少存储块中浪费的空间、重复数据删除可显著提高有效容量。此适用场景数据存储在本机，并分层到云。
- 最低、最高和自适应服务质量(AQoS)。精细的服务质量(QoS)控制有助于在高度共享的环境中保持关键应用程序的性能水平。
- NetApp FabricPool。可将冷数据自动分层到公有和私有云存储选项、包括Amazon Web Services (AWS)、Azure和NetApp StorageGRID Storage解决方案。有关 FabricPool 的详细信息，请参见 ["TR-4598 : FabricPool 最佳实践"](#)。

### 加速和保护数据

ONTAP 可提供卓越的性能和数据保护、并通过以下方式扩展这些功能：

- 性能和更低的延迟。ONTAP 可提供尽可能高的吞吐量和尽可能低的延迟。
- 数据保护ONTAP 可提供内置数据保护功能、并在所有平台之间进行通用管理。
- NetApp卷加密(NVE)。ONTAP 提供原生 卷级加密、并支持板载和外部密钥管理。
- 多租户和多因素身份验证。ONTAP 支持以最高的安全性级别共享基础架构资源。

## Future-Proof 基础架构

ONTAP 可通过以下功能满足不断变化的苛刻业务需求：

- 无缝扩展和无中断运行。ONTAP 支持无中断地向现有控制器和横向扩展集群添加容量。客户可以升级到 NVMe 和 32 Gb FC 等最新技术，而无需进行成本高昂的数据迁移或中断。
- 云连接。ONTAP 是云互联程度最高的存储管理软件、可在所有公有云中选择软件定义的存储(ONTAP Select)和云原生实例(NetApp Cloud Volumes Service)。
- 与新兴应用程序集成。ONTAP 通过使用支持现有企业应用程序的相同基础架构、为下一代平台和应用程序(例如自动驾驶汽车、智能城市和行业4.0)提供企业级数据服务。



## NetApp Astra Control

NetApp Astra 产品系列为内部和公有云中的 Kubernetes 应用程序提供存储和应用程序感知型数据管理服务，并采用 NetApp 存储和数据管理技术。通过它，您可以轻松备份 Kubernetes 应用程序、将数据迁移到其他集群以及即时创建有效的应用程序克隆。如果您需要管理在公有云中运行的 Kubernetes 应用程序，请参见文档 ["Astra 控制服务"](#)。Astra Control Service 是一项由 NetApp 管理的服务，可在 Google Kubernetes Engine（GKEE）和 Azure Kubernetes Service（AKS）中为 Kubernetes 集群提供应用程序感知型数据管理。

## NetApp Astra Trident

Astra ["Trident"](#) NetApp 是适用于 Docker 和 Kubernetes 的开源动态存储编排程序，可简化永久性存储的创建、管理和使用。Trident 是 Kubernetes 本机应用程序，直接在 Kubernetes 集群中运行。借助 Trident，客户可以将 DL 容器映像无缝部署到 NetApp 存储上，并为 AI 容器部署提供企业级体验。Kubernetes 用户（ML 开发人员、数据科学家等）可以创建、管理和自动化流程编排和克隆，从而利用 NetApp 技术提供的高级数据管理功能。

## NetApp BlueXP 复制和同步

["BlueXP 复制和同步"](#) 是一项 NetApp 服务，用于快速安全地同步数据。无论您是需要内部 NFS 还是 SMB 文件共享、NetApp StorageGRID、NetApp ONTAP S3、NetApp Cloud Volumes Service、Azure NetApp Files、Amazon Simple Storage Service (Amazon S3)、Amazon Elastic File System (Amazon EFS)、Azure Blob、Google Cloud Storage、或者 IBM Cloud Object Storage、BlueXP Copy and Sync 可将文件快速安全地移动到您需要的位置。数据传输完成后，即可在源和目标上完全使用。BlueXP Copy 和 Sync 会根据您预定义的计划持续同步数据、仅移动增量、从而最大限度地减少数据复制所需的时间和资金。BlueXP Copy and Sync 是一款软件即服务 (SaaS) 工具，设置和使用极其简单。由 BlueXP 复制和同步触发的数据传输由数据代理执行。您可以在 AWS、Azure、Google Cloud Platform 或内部部署 BlueXP Copy and Sync 数据代理。

## NetApp BlueXP 分类

由强大的 AI 算法驱动，["NetApp BlueXP 分类"](#) 为您的整个数据资产提供自动化控制和数据监管。您可以轻松确定成本节省、发现合规性和隐私问题、并找到优化机会。BlueXP 分类信息板可让您深入了解如何识别重复数据以消除冗余、映射个人数据、非个人数据和敏感数据、以及针对敏感数据和异常情况启用警报。

## 测试和验证计划

在此解决方案设计中，我们对以下三种情形进行了验证：

- 在 JupyterLab 工作空间中使用适用于 Kubernetes 的 NetApp DataOps 工具包进行编排的推理任务，包括使用和不使用 Protopia 混淆。
- 在 Kubernetes 上使用 NetApp DataOps Toolkit for Kubernetes 编排的数据卷进行批量推理作业，包括使用和不使用 Protopia 混淆。
- 使用 NVIDIA Triton 推理服务器实例的推理任务，该实例使用适用于 Kubernetes 的 NetApp DataOps 工具包进行编排。在调用 Triton 推理 API 之前，我们会对映像应用程序混淆，以模拟通过网络传输的任何数据必须被混淆的常见要求。此工作流适用于在受信任区域内收集数据，但必须传递到该受信任区域之外以进行推理的情形。如果没有 Protopia 混淆，则在敏感数据不离开受信任区域的情况下，无法实施此类工作流。

## 测试配置

下表概括了解决方案设计验证环境。

组件	version
Kubernetes	1.21.6
NetApp Astra Trident CSI驱动程序	22.01.0
适用于Kubernetes的NetApp DataOps工具包	2.3.0
NVIDIA Triton 推理服务器	21.11-py3.

## 测试操作步骤

本节介绍完成验证所需的任务。

### 前提条件

要执行本节所述的任务、您必须能够访问安装并配置了以下工具的Linux或macOS主机：

- Kubectl (配置为访问现有Kubernetes集群)
  - 可参见安装和配置说明 ["此处"](#)。
- 适用于Kubernetes的NetApp DataOps工具包
  - 可以找到安装说明 ["此处"](#)。

### 场景1—JupyterLab中的按需推理

1. 为AI/ML推理工作负载创建Kubernetes命名空间。

```
$ kubectl create namespace inference
namespace/inference created
```

2. 使用NetApp DataOps工具包配置永久性卷、以存储要执行推理的数据。

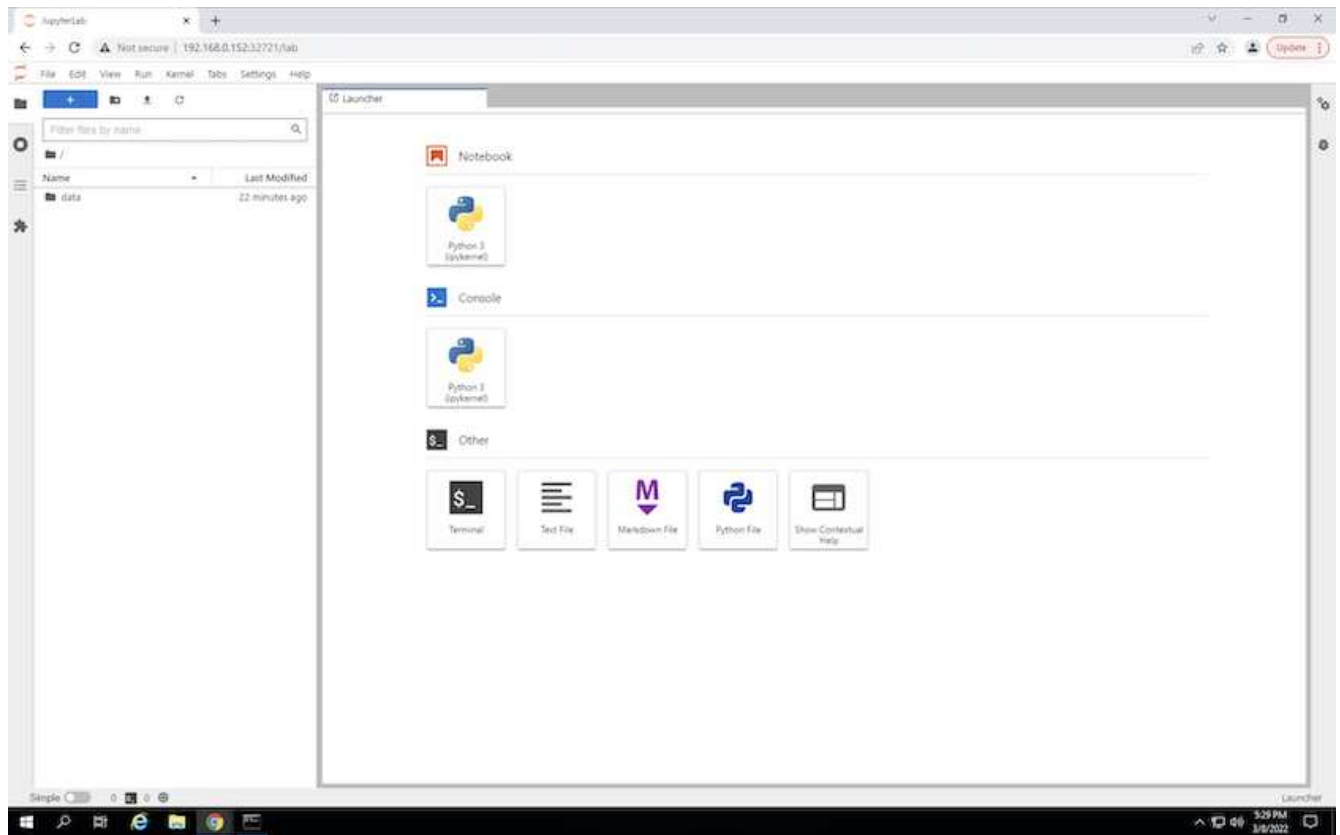
```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'inference-data' in namespace 'inference'.
```

3. 使用NetApp DataOps工具包创建新的JupyterLab工作空间。使用`-mount- PVC`选项挂载上一步中创建的永久性卷。根据需要使用`- nvidia-GPU`选项将NVIDIA GPU分配给工作空间。

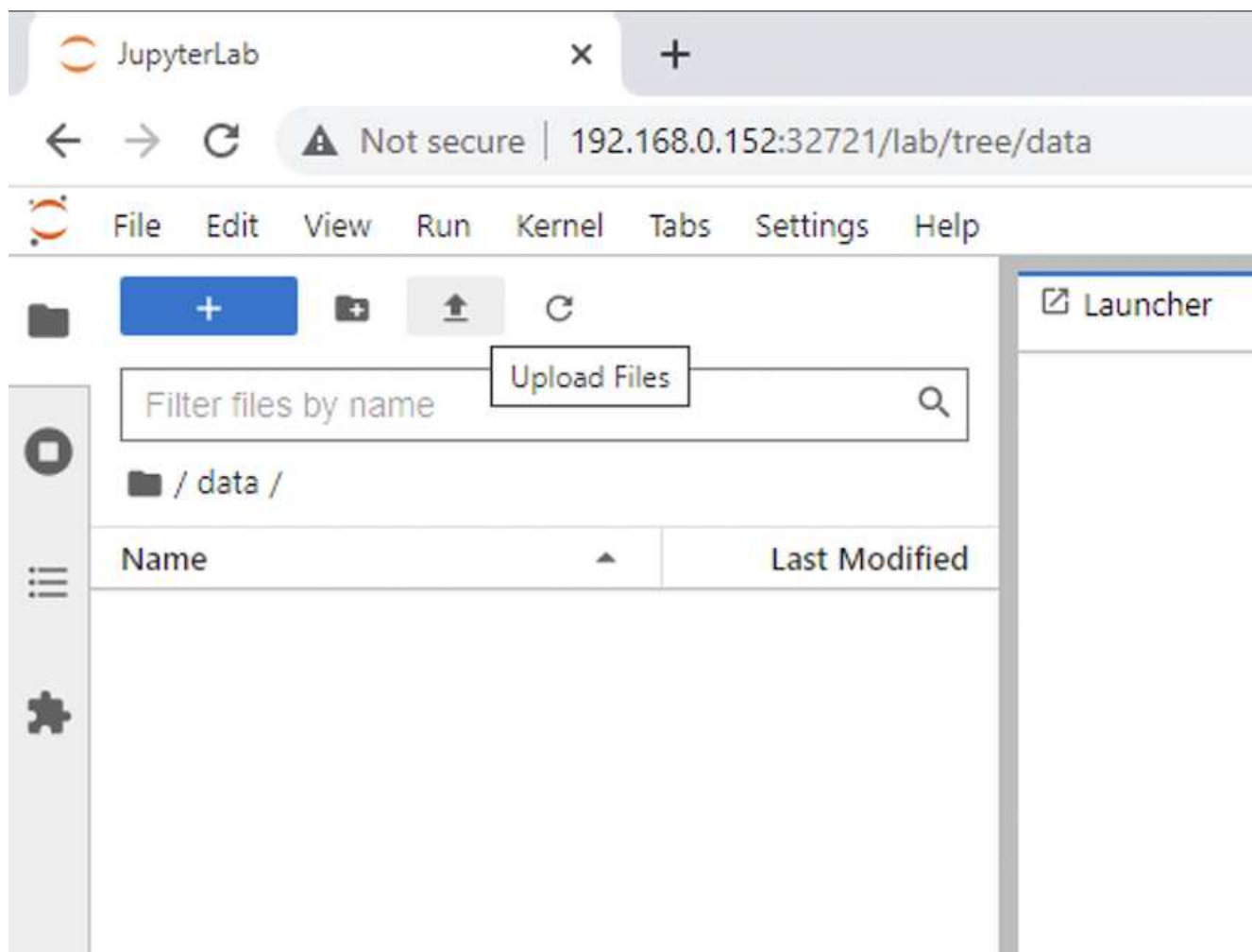
在以下示例中、永久性卷`推理-data`会挂载到JupyterLab工作空间容器中、该容器位于`home/jovyan/data`。使用Project Jupyter官方容器映像时、`/home/jovyan`将作为JupyterLab Web界面中的顶级目录提供。

```
$ netapp_dataops_k8s_cli.py create jupyterlab --namespace=inference
--workspace-name=live-inference --size=50Gi --nvidia-gpu=2 --mount
-pvc=inference-data:/home/jovyan/data
Set workspace password (this password will be required in order to
access the workspace):
Re-enter password:
Creating persistent volume for workspace...
Creating PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-
inference' in namespace 'inference'.
PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-inference'
created. Waiting for Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'ntap-dsutil-jupyterlab-live-inference' in namespace 'inference'.
Creating Service 'ntap-dsutil-jupyterlab-live-inference' in namespace
'inference'.
Service successfully created.
Attaching Additional PVC: 'inference-data' at mount_path:
'/home/jovyan/data'.
Creating Deployment 'ntap-dsutil-jupyterlab-live-inference' in namespace
'inference'.
Deployment 'ntap-dsutil-jupyterlab-live-inference' created.
Waiting for Deployment 'ntap-dsutil-jupyterlab-live-inference' to reach
Ready state.
Deployment successfully created.
Workspace successfully created.
To access workspace, navigate to http://192.168.0.152:32721
```

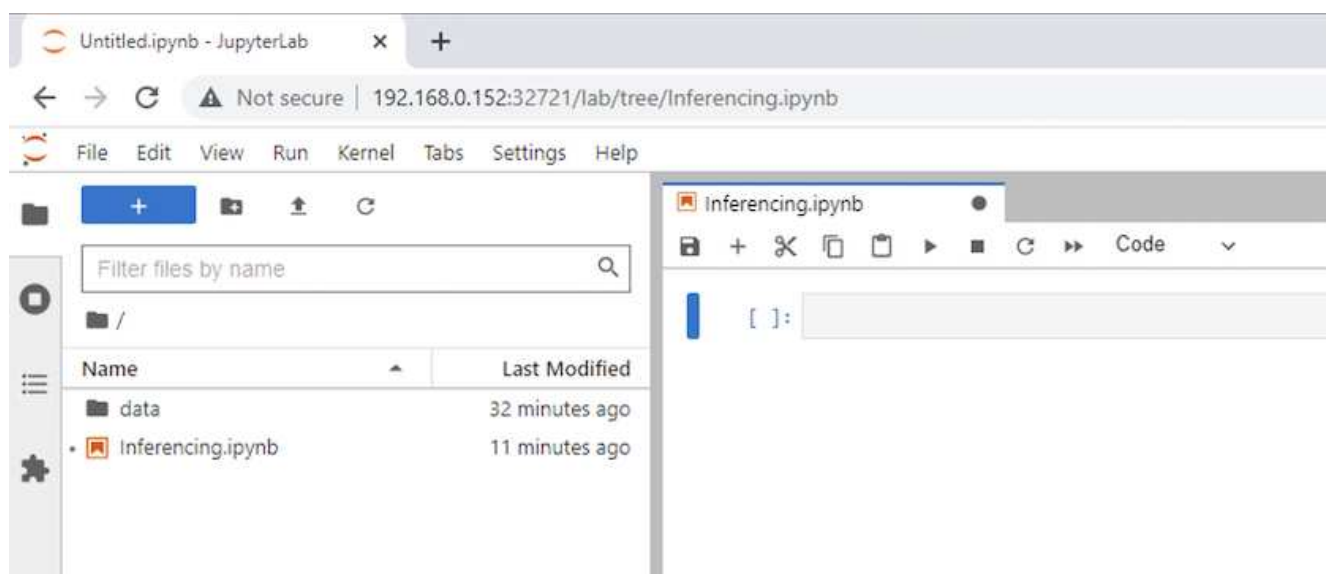
4. 使用`create jupyterlab`命令输出中指定的URL访问JupyterLab工作空间。数据目录表示已挂载到工作空间的永久性卷。



5. 打开`data`目录并上传要执行推理的文件。将文件上传到数据目录时、这些文件会自动存储在挂载到工作空间的永久性卷上。要上传文件、请单击上传文件图标、如下图所示。



6. 返回顶级目录并创建新的笔记本。



7. 向笔记本电脑添加推理代码。以下示例显示了图像检测用例的推理代码。

```
Launcher X image-demo-pytorch.ipynb X Python 3 (ipykernel)

STEP 3-1: Clean (Without obfuscation) detection

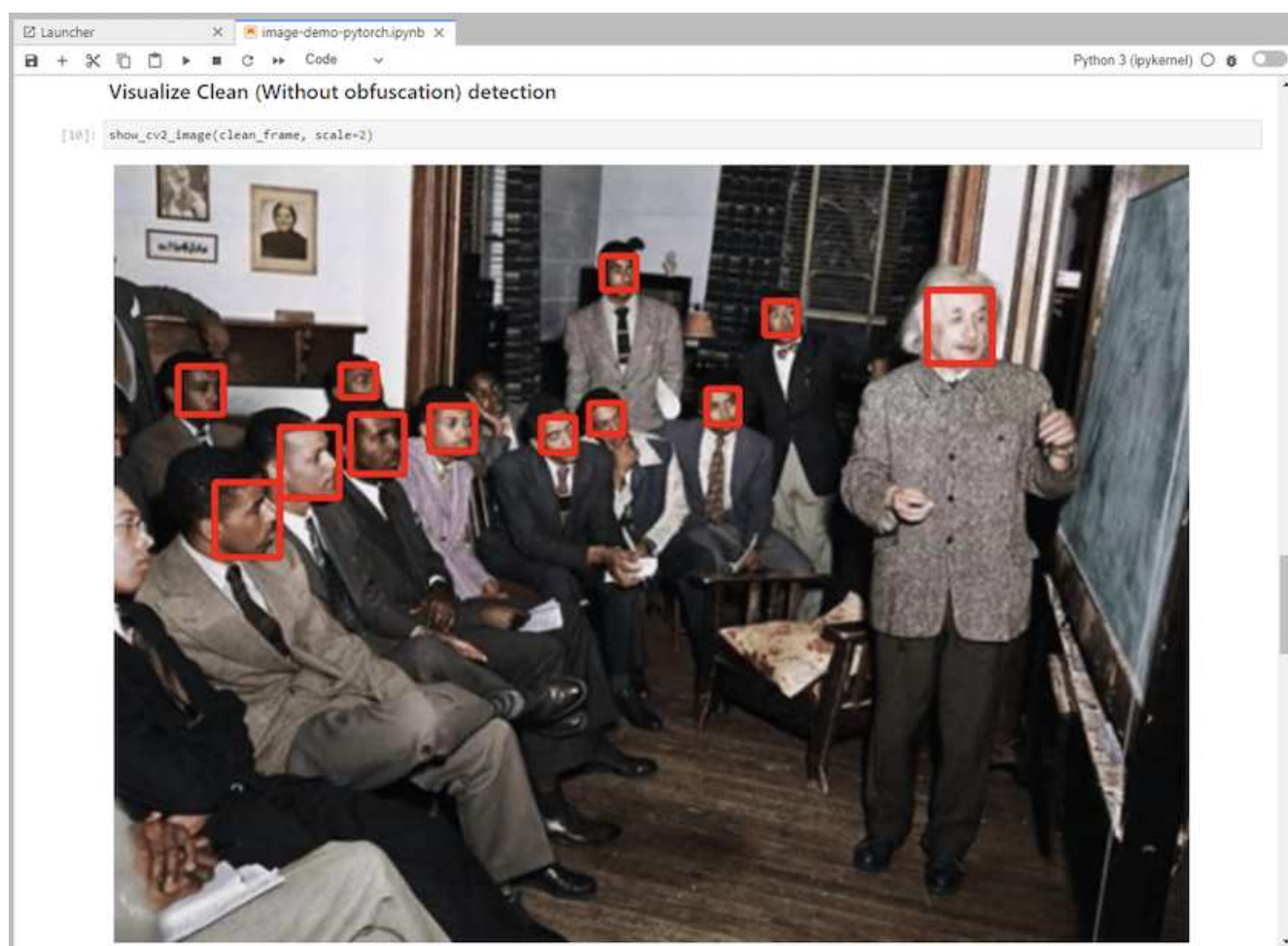
[9]: # get current frame
frame = input_image

# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)

# run forward pass
clean_activation = clean_model.forward_head(preprocessed_input) # runs the first few layers
loc, pred = clean_model.forward_tail(clean_activation) # runs rest of the layers

# postprocess output
clean_pred = (loc.detach().cpu().numpy(), pred.detach().cpu().numpy())
clean_outputs = postprocess_outputs(
    clean_pred, [[input_image_width, input_image_height]], priors, THRESHOLD
)

# draw rectangles
clean_frame = copy.deepcopy(frame) # needs to be deep copy
for (x1, y1, x2, y2, s) in clean_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(clean_frame, (x1, y1), (x2, y2), (0, 0, 255), 4)
```



8. 将Protopia混淆添加到推理代码中。Protopia直接与客户合作、提供特定于使用情形的文档、不在本技术报告的范围。以下示例显示了添加了Protopia混淆的图像检测用例的推理代码。



```
Launcher X image-demo-pytorch.ipynb X Python 3 (ipykernel)

STEP 3-2: Protopia AI (With obfuscation) detection

[11]: # get current frame
      frame = input_image

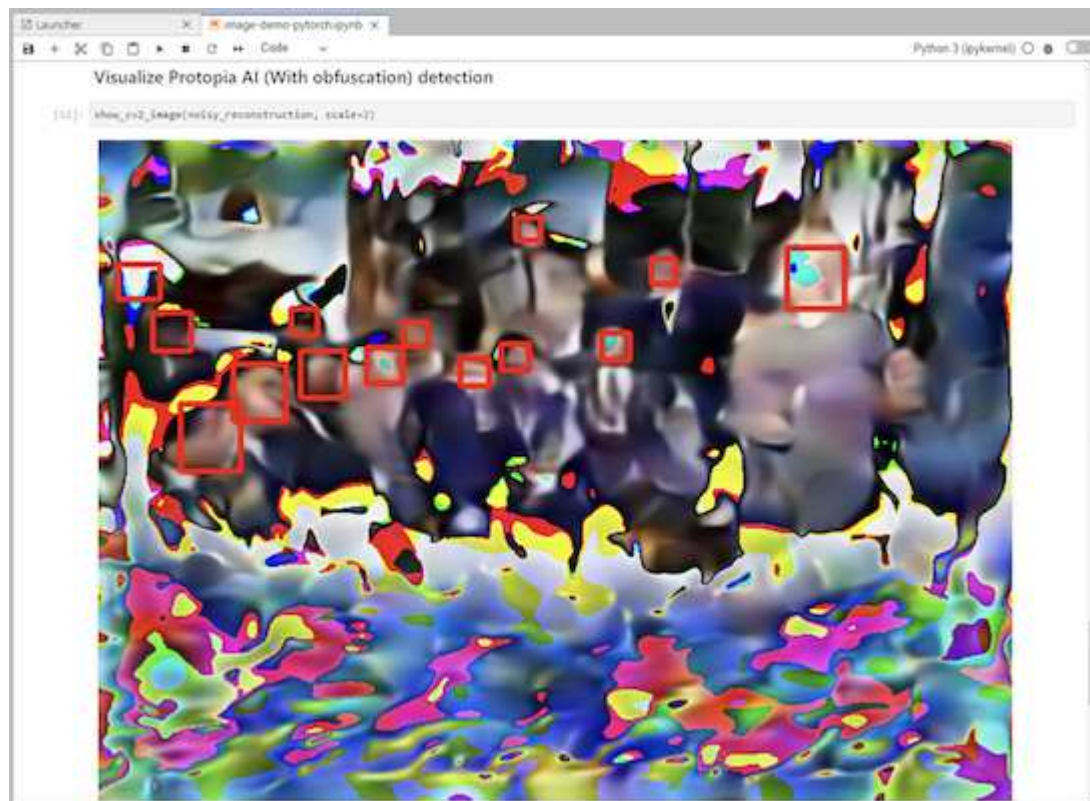
      # preprocess input
      preprocessed_input = preprocess_input(frame)
      preprocessed_input = torch.Tensor(preprocessed_input).to(device)

      # run forward pass
      not_noisy_activation = noisy_model.forward_head(preprocessed_input) # runs the first few layers
      #####
      # SINGLE ADDITIONAL LINE FOR PRIVATE INFERENCE #
      #####
      noisy_activation = noisy_model.forward_noise(not_noisy_activation)
      #####
      loc, pred = noisy_model.forward_tail(noisy_activation) # runs rest of the layers

      # postprocess output
      noisy_pred = (loc.detach().cpu().numpy(), pred.detach().cpu().numpy())
      noisy_outputs = postprocess_outputs(
          noisy_pred, [[input_image_width, input_image_height]], priors, THRESHOLD * 0.5
      )

      # get reconstruction of the noisy activation
      noisy_reconstruction = decoder_function(noisy_activation)
      noisy_reconstruction = noisy_reconstruction.detach().cpu().numpy()[0]
      noisy_reconstruction = unpreprocess_output(
          noisy_reconstruction, (input_image_width, input_image_height), True
      ).astype(np.uint8)

      # draw rectangles
      for (x1, y1, x2, y2, s) in noisy_outputs[0]:
          x1, y1 = int(x1), int(y1)
          x2, y2 = int(x2), int(y2)
          cv2.rectangle(noisy_reconstruction, (x1, y1), (x2, y2), (0, 0, 255), 4)
```



## 场景2—Kubernetes上的批处理推理

1. 为AI/ML推理工作负载创建Kubernetes命名空间。

```
$ kubectl create namespace inference
namespace/inference created
```

2. 使用NetApp DataOps工具包配置永久性卷、以存储要执行推理的数据。

```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'inference-data' in namespace 'inference'.
```

3. 使用要执行推理的数据填充新的永久性卷。

可以通过多种方法将数据加载到PVC上。如果您的数据当前存储在与S3兼容的对象存储平台中、例如NetApp StorageGRID 或Amazon S3、则可以使用 ["NetApp DataOps工具包S3 Data Mover功能"](#)。另一种简单的方法是创建JupyterLab工作空间、然后通过JupyterLab Web界面上传文件、如["一节中的步骤3到5所述场景1—JupyterLab中的按需推理。"](#)

4. 为批处理推理任务创建Kubernetes作业。以下示例显示了一个图像检测用例的批处理推理作业。此作业会对一组映像中的每个映像执行推理、并将推理准确性指标写入到stdout。

```
$ vi inference-job-raw.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-inference-raw
  namespace: inference
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: inference-data
      - name: dshm
        emptyDir:
          medium: Memory
      containers:
      - name: inference
        image:: netapp-protopia-inference:latest
        imagePullPolicy: IfNotPresent
        command: ["python3", "run-accuracy-measurement.py", "--dataset",
"/data/netapp-face-detection/Fddb"]
        resources:
          limits:
            nvidia.com/gpu: 2
        volumeMounts:
        - mountPath: /data
          name: data
        - mountPath: /dev/shm
          name: dshm
        restartPolicy: Never
$ kubectl create -f inference-job-raw.yaml
job.batch/netapp-inference-raw created
```

5. 确认推理作业已成功完成。

```

$ kubectl -n inference logs netapp-inference-raw-255sp
100%|██████████| 89/89 [00:52<00:00, 1.68it/s]
Reading Predictions : 100%|██████████| 10/10 [00:01<00:00, 6.23it/s]
Predicting ... : 100%|██████████| 10/10 [00:16<00:00, 1.64s/it]
===== Results =====
FDDb-fold-1 Val AP: 0.9491256561145955
FDDb-fold-2 Val AP: 0.9205024466101926
FDDb-fold-3 Val AP: 0.9253013871078468
FDDb-fold-4 Val AP: 0.9399781485863011
FDDb-fold-5 Val AP: 0.9504280149478732
FDDb-fold-6 Val AP: 0.9416473519339292
FDDb-fold-7 Val AP: 0.9241631566241117
FDDb-fold-8 Val AP: 0.9072663297546659
FDDb-fold-9 Val AP: 0.9339648715035469
FDDb-fold-10 Val AP: 0.9447707905560152
FDDb Dataset Average AP: 0.9337148153739079
=====
mAP: 0.9337148153739079

```

6. 将Protopia混淆添加到推理作业。您可以从Protopia中找到直接添加Protopia混淆的使用案例专用说明、该说明不在本技术报告的讨论范围内。以下示例显示了一个人脸检测用例的批处理推理作业、该用例使用0.8的字母值添加了质子模糊。此作业会在对一组图像中的每个图像执行推理之前应用程序对象模糊、然后将推理准确性指标写入stdout。

对于alpha值0.05%、0.1、0.2、0.4、0.6、0.8、0.9和0.95。您可以在中查看结果 [""推理准确性比较。""](#)

```

$ vi inference-job-protopia-0.8.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-inference-protopia-0.8
  namespace: inference
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: inference-data
      - name: dshm
        emptyDir:
          medium: Memory
      containers:
      - name: inference
        image:: netapp-protopia-inference:latest
        imagePullPolicy: IfNotPresent
        env:
        - name: ALPHA
          value: "0.8"
        command: ["python3", "run-accuracy-measurement.py", "--dataset",
"/data/netapp-face-detection/Fddb", "--alpha", "$(ALPHA)", "--noisy"]
        resources:
          limits:
            nvidia.com/gpu: 2
        volumeMounts:
        - mountPath: /data
          name: data
        - mountPath: /dev/shm
          name: dshm
        restartPolicy: Never
$ kubectl create -f inference-job-protopia-0.8.yaml
job.batch/netapp-inference-protopia-0.8 created

```

## 7. 确认推理作业已成功完成。

```
$ kubectl -n inference logs netapp-inference-protopia-0.8-b4dkz
100%|██████████| 89/89 [01:05<00:00, 1.37it/s]
Reading Predictions : 100%|██████████| 10/10 [00:02<00:00, 3.67it/s]
Predicting ... : 100%|██████████| 10/10 [00:22<00:00, 2.24s/it]
===== Results =====
FDDB-fold-1 Val AP: 0.8953066115834589
FDDB-fold-2 Val AP: 0.8819580264029936
FDDB-fold-3 Val AP: 0.8781107458462862
FDDB-fold-4 Val AP: 0.9085731346308461
FDDB-fold-5 Val AP: 0.9166445508275378
FDDB-fold-6 Val AP: 0.9101178994188819
FDDB-fold-7 Val AP: 0.8383443678423771
FDDB-fold-8 Val AP: 0.8476311547659464
FDDB-fold-9 Val AP: 0.8739624502111121
FDDB-fold-10 Val AP: 0.8905468076424851
FDDB Dataset Average AP: 0.8841195749171925
=====
mAP: 0.8841195749171925
```

### 场景3—NVIDIA Triton推理服务器

1. 为AI/ML推理工作负载创建Kubernetes命名空间。

```
$ kubectl create namespace inference
namespace/inference created
```

2. 使用NetApp DataOps工具包配置永久性卷、以用作NVIDIA Triton推理服务器的型号存储库。

```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=triton-model-repo --size=100Gi
Creating PersistentVolumeClaim (PVC) 'triton-model-repo' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'triton-model-repo' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'triton-model-repo' in namespace 'inference'.
```

3. 将您的型号存储在中的新永久性卷上“格式。”这可由NVIDIA Triton推理服务器识别。

可以通过多种方法将数据加载到PVC上。一种简单的方法是创建JupyterLab工作空间、然后通过JupyterLab Web界面上传文件、如中的步骤3到5所述[场景1—JupyterLab中的按需推理](#)。”

4. 使用NetApp DataOps工具包部署新的NVIDIA Triton推理服务器实例。



```

$ netapp_dataops_k8s_cli.py create triton-server --namespace=inference
--server-name=netapp-inference --model-repo-pvc-name=triton-model-repo
Creating Service 'ntap-dsutil-triton-netapp-inference' in namespace
'inference'.
Service successfully created.
Creating Deployment 'ntap-dsutil-triton-netapp-inference' in namespace
'inference'.
Deployment 'ntap-dsutil-triton-netapp-inference' created.
Waiting for Deployment 'ntap-dsutil-triton-netapp-inference' to reach
Ready state.
Deployment successfully created.
Server successfully created.
Server endpoints:
http: 192.168.0.152: 31208
grpc: 192.168.0.152: 32736
metrics: 192.168.0.152: 30009/metrics

```

5. 使用Triton客户端SDK执行推理任务。以下Python代码摘录使用Triton Python客户端SDK为人脸检测用例执行推理任务。此示例调用Triton API并传递图像以进行推理。然后、Triton推理服务器接收请求、调用模型、并在API结果中返回推理输出。

```

# get current frame
frame = input_image
# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)
# run forward pass
clean_activation = clean_model_head(preprocessed_input) # runs the
first few layers
#####
#####
#           pass clean image to Triton Inference Server API for
inferencing           #
#####
#####
triton_client =
httpclient.InferenceServerClient(url="192.168.0.152:31208",
verbose=False)
model_name = "face_detection_base"
inputs = []
outputs = []
inputs.append(httpclient.InferInput("INPUT__0", [1, 128, 32, 32],
"FP32"))
inputs[0].set_data_from_numpy(clean_activation.detach().cpu().numpy(),
binary_data=False)

```

```

outputs.append(httpclient.InferRequestedOutput("OUTPUT__0",
binary_data=False))
outputs.append(httpclient.InferRequestedOutput("OUTPUT__1",
binary_data=False))
results = triton_client.infer(
    model_name,
    inputs,
    outputs=outputs,
    #query_params=query_params,
    headers=None,
    request_compression_algorithm=None,
    response_compression_algorithm=None)
#print(results.get_response())
statistics =
triton_client.get_inference_statistics(model_name=model_name,
headers=None)
print(statistics)
if len(statistics["model_stats"]) != 1:
    print("FAILED: Inference Statistics")
    sys.exit(1)

loc_numpy = results.as_numpy("OUTPUT__0")
pred_numpy = results.as_numpy("OUTPUT__1")
#####
#####
# postprocess output
clean_pred = (loc_numpy, pred_numpy)
clean_outputs = postprocess_outputs(
    clean_pred, [[input_image_width, input_image_height]], priors,
    THRESHOLD
)
# draw rectangles
clean_frame = copy.deepcopy(frame) # needs to be deep copy
for (x1, y1, x2, y2, s) in clean_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(clean_frame, (x1, y1), (x2, y2), (0, 0, 255), 4)

```

6. 将Protopia混淆添加到推理代码中。您可以从Protopia中找到直接添加Protopia混淆的使用案例专用说明；但是、此过程不在本技术报告的讨论范围内。以下示例显示了与上一步5中显示的相同的Python代码、但添加了Protopia obfuscation。

请注意、在将图像传递到Triton API之前、系统会对该映像应用程序模糊。因此、非混淆映像永远不会离开本地计算机。仅通过网络传递模糊映像。此 workflow 适用于以下情形：在受信任区域内收集数据、但随后需要传递到该受信任区域以外以进行推理。如果没有Protopia混淆、则在敏感数据不离开受信任区域的情况下、无法实施此类 workflow。

```

# get current frame
frame = input_image
# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)
# run forward pass
not_noisy_activation = noisy_model_head(preprocessed_input) # runs the
first few layers
#####
#           obfuscate image locally prior to inferencing           #
#           SINGLE ADITIONAL LINE FOR PRIVATE INFERENCE           #
#####
noisy_activation = noisy_model_noise(not_noisy_activation)
#####
#####
#####
#           pass obfuscated image to Triton Inference Server API for
inferencing           #
#####
#####
triton_client =
httpclient.InferenceServerClient(url="192.168.0.152:31208",
verbose=False)
model_name = "face_detection_noisy"
inputs = []
outputs = []
inputs.append(httpclient.InferInput("INPUT__0", [1, 128, 32, 32],
"FP32"))
inputs[0].set_data_from_numpy(noisy_activation.detach().cpu().numpy(),
binary_data=False)
outputs.append(httpclient.InferRequestedOutput("OUTPUT__0",
binary_data=False))
outputs.append(httpclient.InferRequestedOutput("OUTPUT__1",
binary_data=False))
results = triton_client.infer(
    model_name,
    inputs,
    outputs=outputs,
    #query_params=query_params,
    headers=None,
    request_compression_algorithm=None,
    response_compression_algorithm=None)
#print(results.get_response())
statistics =
triton_client.get_inference_statistics(model_name=model_name,
headers=None)

```

```

print(statistics)
if len(statistics["model_stats"]) != 1:
    print("FAILED: Inference Statistics")
    sys.exit(1)

loc_numpy = results.as_numpy("OUTPUT__0")
pred_numpy = results.as_numpy("OUTPUT__1")
#####

#####

# postprocess output
noisy_pred = (loc_numpy, pred_numpy)
noisy_outputs = postprocess_outputs(
    noisy_pred, [[input_image_width, input_image_height]], priors,
    THRESHOLD * 0.5
)
# get reconstruction of the noisy activation
noisy_reconstruction = decoder_function(noisy_activation)
noisy_reconstruction = noisy_reconstruction.detach().cpu().numpy()[0]
noisy_reconstruction = unpreprocess_output(
    noisy_reconstruction, (input_image_width, input_image_height), True
).astype(np.uint8)
# draw rectangles
for (x1, y1, x2, y2, s) in noisy_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(noisy_reconstruction, (x1, y1), (x2, y2), (0, 0, 255),
4)

```

## 推理准确性比较

在此验证中、我们使用一组原始映像为图像检测用例执行推理。然后、我们对同一组图像执行了相同的推理任务、并在推理之前添加了质子模糊。我们对Protopia混淆组件使用不同的alpha值重复执行此任务。在Protopia混淆的上下文中、alpha值表示应用的混淆量、较高的alpha值表示较高的混淆级别。然后、我们对这些不同的运行中的推理准确性进行了比较。

以下两个表提供了有关我们使用情形的详细信息并概述了结果。

Protopia直接与客户合作、为特定使用情形确定适当的alpha值。

组件	详细信息
型号	面板(PyTorch)-
数据集	FDDB数据集

前倾混淆	字母	准确性
否	不适用	0.9337148153739079
是的。	0.05	0.9028766627325002
是的。	0.1	0.9024301009661478
是的。	0.2	0.9081836283186224
是的。	0.4	0.9073066107482036
是的。	0.6	0.8847816568680239
是的。	0.8	0.8841195749171925
是的。	0.9	0.8455427675252052
是的。	0.95	0.8455427675252052

## 模糊速度

在此验证中、我们对1920 x 1080像素图像应用了五次Protopia混淆、并测量了每次完成混淆步骤所需的时间。

我们使用在一个NVIDIA V100 GPU上运行的PyTorch来应用混淆、并在两次运行之间清除了GPU缓存。在五次运行中、模糊步骤分别需要5.47毫秒、5.27毫秒、4.54毫秒、5.24毫秒和4.84毫秒才能完成。平均速度为5.072毫秒。

## 结论

数据有三种状态：空闲、传输和计算。任何AI推理服务的一个重要部分都应该是在整个过程中保护数据免受威胁。在推理期间保护数据至关重要、因为该过程可以公开有关外部客户和提供推理服务的企业的私有信息。Protopia AI是一种在当今市场中用于机密人工智能推理的纯软件解决方案。借助Protopia、AI只会从数据记录中传输经过转换的信息、而这些信息对于执行手头的AI/ML任务至关重要、而且不会再提供更多信息。这种随机转型不是一种屏蔽形式、而是基于使用精心设计的噪声以数学方式更改数据表示形式。

具有ONTAP 功能的NetApp存储系统可提供与本地SSD存储相同或更好的性能、并且与NetApp DataOps工具包相结合、可为数据科学家、数据工程师、AI/ML开发人员以及业务或企业IT决策者带来以下优势：

- 在 AI 系统，分析和其他关键业务系统之间轻松共享数据。这种数据共享可减少基础架构开销，提高性能并简化整个企业的数据管理。
- 可独立扩展的计算和存储，最大限度地降低成本并提高资源利用率。
- 利用集成的 Snapshot 副本和克隆简化开发和部署工作流，实现瞬时且节省空间的用户工作空间，集成版本控制和自动化部署。
- 针对灾难恢复、业务连续性和法规要求的企业级数据保护和数据监管。
- 简化了数据管理操作的调用；从Jupyter笔记本电脑中的NetApp DataOps工具包快速获取数据科学家工作空间的Snapshot副本、以实现备份和可追溯性。

NetApp和Protopia解决方案 提供了一个灵活的横向扩展架构、非常适合企业级AI推理部署。它支持数据保护、并为敏感信息提供隐私保护、可通过内部部署和混合云部署中的负责AI实践满足机密AI推理要求。

## 从何处查找追加信息和相关信息

要了解有关本文档中所述信息的更多信息，请参见以下文档和 / 或网站：

- NetApp ONTAP 数据管理软件—ONTAP 信息库

<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>

- 适用于容器的 NetApp 持久存储— NetApp Trident

["https://netapp.io/persistent-storage-provisioner-for-kubernetes/"](https://netapp.io/persistent-storage-provisioner-for-kubernetes/)

- NetApp DataOps 工具包

["https://github.com/NetApp/netapp-dataops-toolkit"](https://github.com/NetApp/netapp-dataops-toolkit)

- 适用于容器的NetApp持久存储—NetApp Astra Trident

["https://netapp.io/persistent-storage-provisioner-for-kubernetes/"](https://netapp.io/persistent-storage-provisioner-for-kubernetes/)

- Protopia AI—机密推理

["https://protopia.ai/blog/protopia-ai-takes-on-the-missing-link-in-ai-privacy-confidential-inference/"](https://protopia.ai/blog/protopia-ai-takes-on-the-missing-link-in-ai-privacy-confidential-inference/)

- NetApp BlueXP复制和同步

["https://docs.netapp.com/us-en/occm/concept\\_cloud\\_sync.html#how-cloud-sync-works"](https://docs.netapp.com/us-en/occm/concept_cloud_sync.html#how-cloud-sync-works)

- NVIDIA Triton 推理服务器

["https://developer.nvidia.com/nvidia-triton-inference-server"](https://developer.nvidia.com/nvidia-triton-inference-server)

- NVIDIA Triton推理服务器文档

["https://docs.nvidia.com/deeplearning/triton-inference-server/index.html"](https://docs.nvidia.com/deeplearning/triton-inference-server/index.html)

- PyTorch中的FaceBoxes

["https://github.com/zisianw/FaceBoxes.PyTorch"](https://github.com/zisianw/FaceBoxes.PyTorch)

## 致谢

- NetApp首席产品经理Mark Cates
- NetApp技术营销工程师Sufian Ahmad
- 首席技术官兼Protopia AI教授Hadi Essmaeilzadeh

## 使用 NetApp AI 进行情感分析



## TR-4910：《利用 NetApp AI 进行客户沟通时的情感分析》

SML 科学公司 Rick Huang，Sathish Thyagarajan 和 NetApp 公司迭戈 Sosa-Coba 的 David Arnette

本技术报告为客户在企业级全球支持中心内使用 NetApp 数据管理技术和 NVIDIA 软件框架以及传输学习和对话 AI 执行情感分析提供了设计指导。此解决方案适用于希望从代表聊天日志，电子邮件和其他文本或音频通信的录制语音或文本文件中获得客户见解的任何行业。我们实施了一个端到端管道，用于在采用 NetApp 云连接全闪存存储的 GPU 加速计算集群上展示自动语音识别，实时情感分析和深度学习的自然语言处理模式 - 再培训功能。可以对大规模的一流语言模型进行培训和优化，以便与全球支持中心快速进行推理，从而打造卓越的客户体验，并对员工的长期业绩进行客观评估。

情绪分析是自然语言处理（NLP）中的一个研究领域，通过该领域，可以从文本中提取积极，负面或中立的情绪。随着越来越多的人与人工智能系统互动，对话式人工智能系统的集成程度已提升到接近全球水平。情感分析有多种使用情形，从确定支持中心员工在与来电者对话中的表现，提供适当的自动聊天机器人响应，到根据公司代表与受众在季度收益通话中的互动预测公司的股票价格。此外，情感分析可用于确定客户对品牌提供的产品，服务或支持的看法。

此端到端解决方案使用 NLP 模型执行高级别的情感分析，从而支持支持中心分析框架。录音会处理成书面文本，并从对话的每一句话中提取情感。可以将结果汇总到信息板中，以便对对话感受进行分析，包括历史和实时分析。此解决方案可以概括为具有类似数据模式和输出需求的其他解决方案。利用适当的数据，可以完成其他使用情形。例如，可以使用相同的端到端管道分析公司收益调用的情绪。由于管道的灵活性，还可以进行其他形式的 NLP 分析，例如主题建模和命名实体识别（NER）。

NVIDIA Riva，NVIDIA TAO 工具包和 NetApp DataOps 工具包共同实现了这些 AI 实施。NVIDIA 的工具可用于使用预构建的型号和管道快速部署高性能 AI 解决方案。NetApp DataOps 工具包可简化各种数据管理任务，以加快开发速度。

### 客户价值

企业可以通过员工评估和客户响应工具在文本，音频和视频对话中获得价值，以便进行情绪分析。经理可以从信息板中提供的信息中受益，从而可以根据对话的两个方面评估员工和客户满意度。

此外，NetApp DataOps 工具包还可管理客户基础架构中数据的版本控制和分配。这样就会频繁更新信息板中提供的分析，而不会产生庞大的数据存储成本。

### 用例

由于这些支持中心处理的呼叫数量众多，如果手动执行，则可能需要很长时间才能评估呼叫性能。传统方法，例如字包计数和其他方法，可以实现一定程度的自动化，但这些方法不能捕获动态语言的更细微的方面和语义上下文。AI 建模技术可用于自动执行其中一些更精细的分析。此外，随着 NVIDIA，AWS，Google 等公司发布的最新的经过预先培训的建模工具，现在可以相对轻松地建立一个包含复杂模型的端到端管道并进行自定义。

支持中心情感分析的端到端管道，可在员工与来电者交谈时实时载入音频文件。然后，这些音频文件将进行处理，以便在将其转换为文本格式的语音到文本组件中使用。对话中的每一句话都会收到一个标签，用于指示情绪（积极，消极或中立）。

情绪分析可以为对话提供一个重要方面，用于评估通话性能。这些情感为员工和来电者之间的互动增加了一个深度。人工智能辅助的情绪信息板可为经理实时跟踪对话中的情绪，并对员工过去的通话进行回顾性分析。

预构建的工具可以通过强大的方式组合在一起，快速创建端到端 AI 管道来解决此问题。在这种情况下，可以使用 NVIDIA Riva 库执行两项串联任务：音频抄写和情感分析。第一种是受监控学习信号处理算法，第二种是受监控学习 NLP 分类算法。这些即装即用的算法可以使用 NVIDIA TAO 工具包针对与业务相关的数据的任何相关使用情形进行微调。这样，构建的解决方案就会更加准确，功能更强大，只需成本和资源的一小部分。客户可以采用 "NVIDIA Maxine" 支持中心设计中 GPU 加速视频会议应用程序的框架。

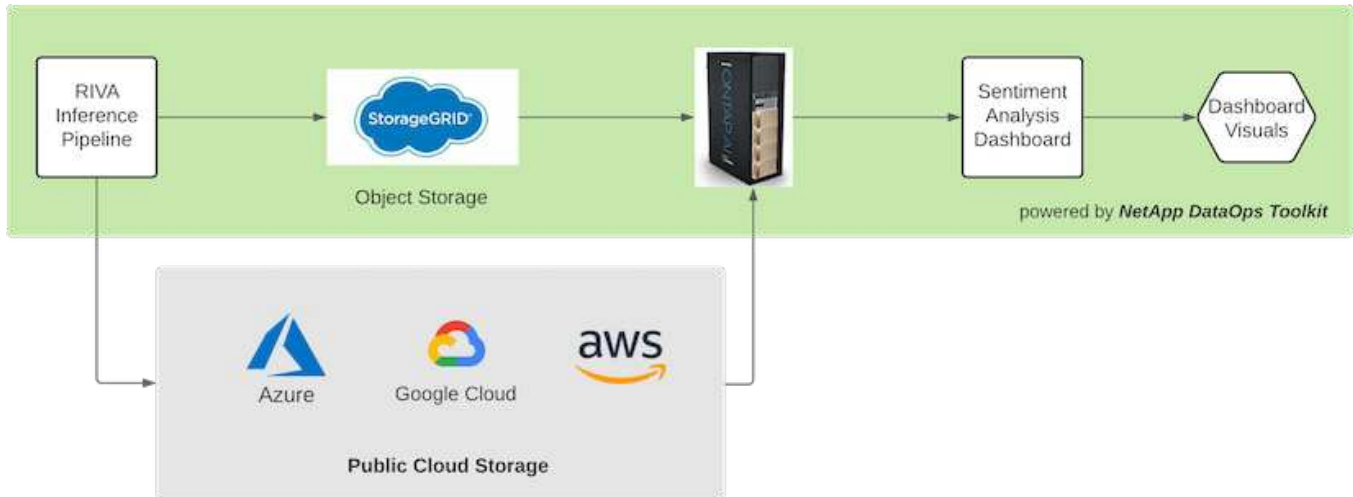
以下用例是此解决方案的核心。这两种使用情形都使用 TAO 工具包进行模型微调，使用 Riva 进行模型部署。

- 语音到文本
- 情感分析

为了分析员工和客户之间的支持中心互动，可以通过管道进行每一次以音频通话形式进行的客户对话，以得出句子级别的感受。然后，人类可以验证这些情感，为这些情感提供合理的理由，或者根据需要进行调整。然后，标记的数据将传递到微调步骤，以改善情绪预测。如果已存在标记的情感数据，则可以加快模型微调。无论哪种情况，管道都可通用于需要输入音频和对句子进行分类的其他解决方案。



AI Sentiment 输出会上传到外部云数据库或公司管理的存储系统。情感输出将从这一更大的数据库传输到本地存储，以在为经理显示情感分析的信息板中使用。信息板的主要功能是与客户服务员工实时互动。经理可以通过实时更新每句话的感受，并对员工过去的表现或客户反应进行历史回顾，评估员工的通话并提供反馈。



。"NetApp DataOps 工具包" 即使在 Riva 推理管道生成情感标签之后，也可以继续管理数据存储系统。这些 AI 结果可以上传到 NetApp DataOps 工具包管理的数据存储系统。数据存储系统必须能够管理数百个插件并每分钟进行一次选择。本地设备存储系统实时查询较大的数据存储以进行提取。此外，还可以查询较大的数据存储实例以查找历史数据，从而进一步增强信息板体验。NetApp DataOps 工具包可通过快速克隆数据并将其分发到使用该数据的所有信息板，从而为这两种用途提供便利。

## 目标受众

解决方案的目标受众包括以下组：

- 员工经理
- 数据工程师 / 数据科学家
- IT 管理员（内部，云或混合）

在整个对话过程中跟踪情绪是评估员工绩效的宝贵工具。通过使用 AI 信息板，经理可以实时了解员工和来电者如何改变自己的感受，从而可以开展实时评估和指导会议。此外，企业还可以从参与语音对话，文本聊天机器人和视频会议的客户中获得宝贵的客户见解。此类客户分析可利用大规模多模式处理功能以及现代化的一流 AI 模型和工作流。

在数据方面，支持中心每天都会处理大量音频文件。NetApp DataOps 工具包可帮助执行此数据处理任务，以便定期对模型和情感分析信息板进行微调。

IT 管理员还可以从 NetApp DataOps 工具包中受益，因为它允许他们在部署和生产环境之间快速移动数据。此外，还必须对 NVIDIA 环境和服务器进行管理和分布，以便进行实时推理。

## 架构

此支持中心解决方案的架构以 NVIDIA 的预构建工具和 NetApp DataOps 工具包为核心。NVIDIA 的工具可用于使用预构建的型号和管道快速部署高性能 AI 解决方案。NetApp DataOps 工具包可简化各种数据管理任务，以加快开发速度。

## 解决方案技术

"NVIDIA RIVA" 是一款 GPU 加速 SDK，用于构建多模式对话 AI 应用程序，在 GPU 上提供实时性能。NVIDIA

Train , Aadapt , and Optimize （TAO）工具包提供了一种更快，更简单的方法来加快训练速度，并快速创建高度准确且性能优异的特定域 AI 模型。

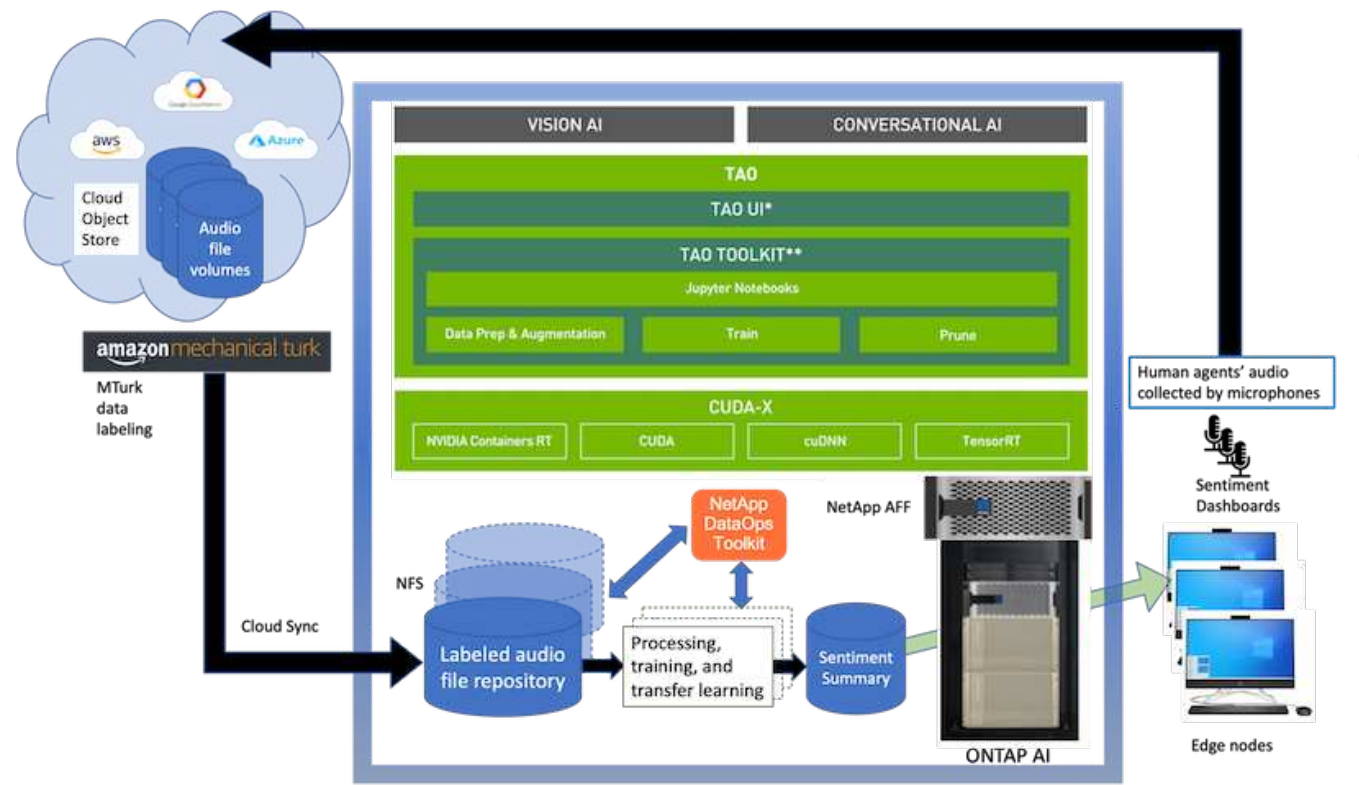
NetApp DataOps 工具包是一个 Python 库，可使开发人员，数据科学家，开发运营工程师和数据工程师轻松执行各种数据管理任务。其中包括近乎即时地配置新的数据卷或 JupyterLab 工作空间，近乎即时地克隆数据卷或 JupyterLab 工作空间，以及近乎即时地对数据卷或 JupyterLab 工作空间进行快照，以实现可追溯性和基线化。

架构图

下图显示了解决方案架构。环境分为三大类：云，核心和边缘。每个类别都可能在不同的地理位置上分散。例如、云包含的对象存储在不同区域的分段中包含音频文件、而核心可能包含通过高速网络或NetApp BlueXP复制和同步链接的数据中心。边缘节点表示各个人工代理的日常工作平台，在这些平台中，交互式信息板工具和麦克风可用于直观显示情绪并从与客户的对话中收集音频数据。

在 GPU 加速数据中心中，企业可以使用 NVIDIA "Riva" 用于构建对话式 AI 应用程序的框架 "TAO 工具包" 使用传输 L 学习技术连接模型修剪和重新培训。这些计算应用程序和工作流由提供支持 "NetApp DataOps 工具包"，实现 ONTAP 提供的最佳数据管理功能。通过该工具包，企业数据团队可以利用相关的结构化和非结构化数据，通过快照和克隆快速构建模型的原型，以实现可追溯性，版本控制，A/B 测试，从而提供安全性，监管，和合规性。请参见一节 "存储设计" 有关详细信息：

此解决方案演示了音频文件处理， NLP 模型培训，传输学习和数据管理详细步骤。生成的端到端管道生成一个情感摘要，该摘要会实时显示在人力支持代理的信息板上。



硬件要求

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

响应延迟测试	时间（毫秒）
数据处理	10
推理	10

这些响应时间测试针对 560 个对话中 50,000 多个音频文件运行。每个音频文件的大小为 ~100KB，转换为 wav 时大小为 ~1 MB。数据处理步骤会将 MP3 转换为 wav 文件。推理步骤会将音频文件转换为文本，并从文本中提取感受。这些步骤彼此独立，可以并行处理以加快此过程。

考虑到在商店之间传输数据的延迟，经理应该能够在句子结尾的一秒内看到实时情感分析的更新。

## NVIDIA Riva 硬件

硬件	要求
OS	Linux x86_64
GPU 内存（As1）	流模型：~5600 MB 非流模型：~3100 MB
GPU 内存（NLP）	~每个 Bert 型号 500 MB

## NVIDIA TAO 工具包硬件

硬件	要求
系统 RAM	32 GB
GPU RAM	32 GB
CPU	8 个核心
GPU	NVIDIA（A100，V100 和 RTX 30x0）
SSD	100 GB

## 闪存存储系统

## NetApp ONTAP 9.

ONTAP 9.9 是 NetApp 推出的最新一代存储管理软件，可帮助企业打造现代化的基础架构并过渡到云就绪数据中心。借助行业领先的数据管理功能，无论数据位于何处，ONTAP 都可以通过一组工具来管理和保护数据。您还可以将数据自由移动到需要的任何位置：边缘，核心或云。ONTAP 9.9 包含许多功能，可简化数据管理，加快和保护关键数据，并在混合云架构中实现下一代基础架构功能。

## NetApp BlueXP 复制和同步

"BlueXP 复制和同步" 是一项 NetApp 服务，用于快速安全地同步数据，使您可以在内部 NFS 或 SMB 文件共享之间将文件传输到以下任一目标：

- NetApp StorageGRID
- NetApp ONTAP S3
- NetApp Cloud Volumes Service
- Azure NetApp Files

- Amazon Simple Storage Service （ Amazon S3 ）
- Amazon Elastic File System （ Amazon EFS ）
- Azure Blob
- Google Cloud 存储
- IBM 云对象存储

BlueXP复制和同步功能可将文件快速安全地移动到您需要的位置。数据传输完成后，即可在源和目标上完全使用。BlueXP复制和同步功能会根据预定义的计划持续同步数据、仅移动增量、从而最大限度地减少数据复制所需的时间和资金。BlueXP Copy and Sync是一款软件即服务(SaaS)工具、易于设置和使用。由BlueXP复制和同步触发的数据传输由数据代理执行。您可以在AWS、Azure、Google Cloud Platform或内部部署BlueXP Copy and Sync数据代理。

## NetApp StorageGRID

StorageGRID 软件定义的对象存储套件可无缝支持公有，私有云和混合多云环境中的各种用例。借助行业领先的创新技术， NetApp StorageGRID 可存储，保护和保留非结构化数据，以供多用途使用，包括长期的自动化生命周期管理。有关详细信息，请参见 "[NetApp StorageGRID](#)" 站点

### 软件要求

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

主机	要求
Riva （原 JarVis ）	1.4.0
TAO 工具包（以前称为传输学习工具包）	3.0
ONTAP	9.9.1
DGX 操作系统	5.1
DOTK	2.0.0

## NVIDIA Riva 软件

软件	要求
Docker	如果不使用 DGX ， 则大于 19.02 （安装了 NVIDIA-Docker ） >=19.03
NVIDIA 驱动程序	对于数据中心 GPU ， 则为 465.19.01+ 418.40+ ， 440.33+ ， 450.51+ ， 460.27+
容器操作系统	Ubuntu 20.04
CUDA	11.3.0
cucBLAS	11.5.1.101
cuDNN	8.2.0.41
NCCL	2.9.6
TensorRT	7.2.3.4



软件	要求
Triton 推理服务器	2.9.0

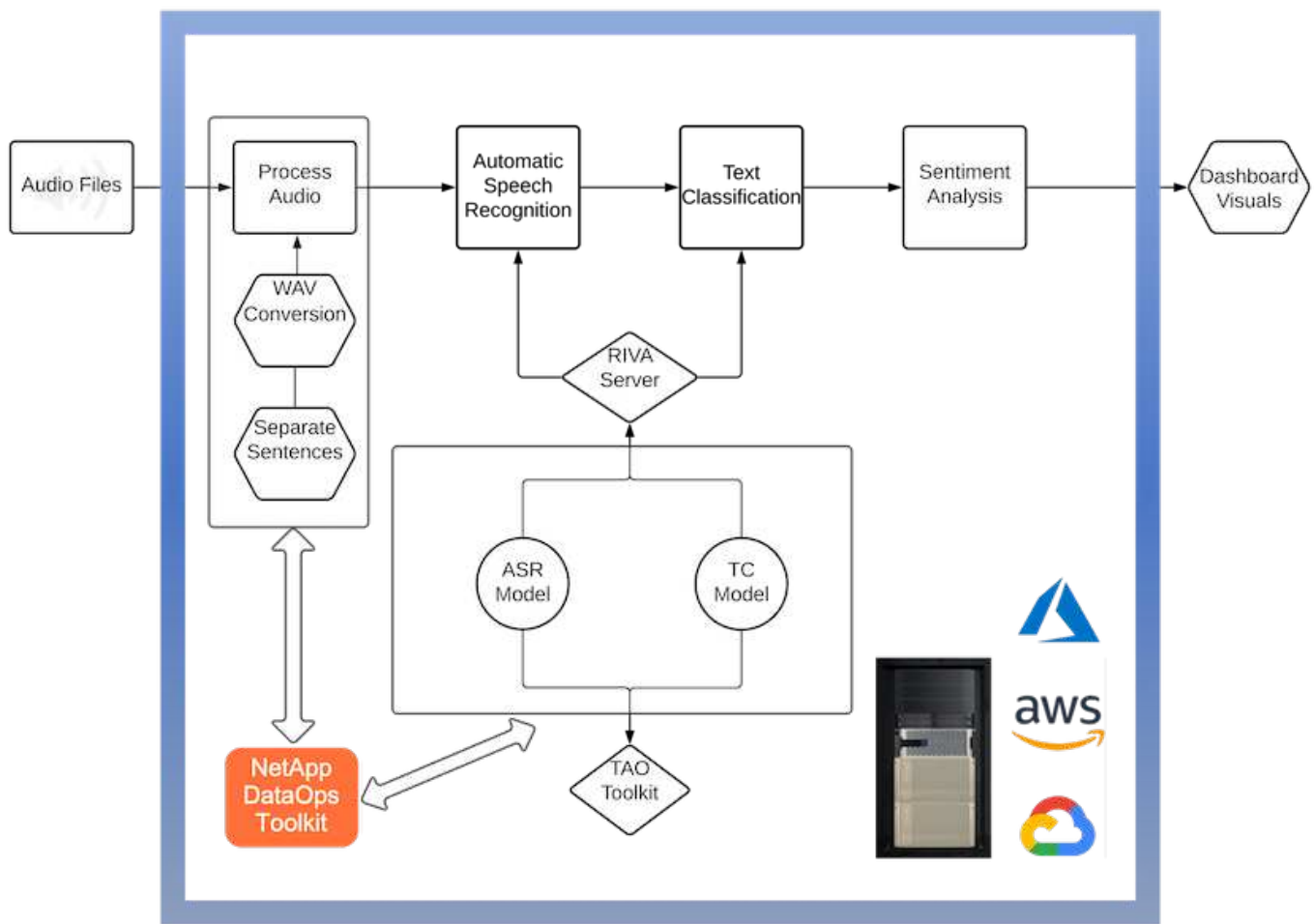
## NVIDIA TAO 工具包软件

软件	要求
Ubuntu 18.04 LTS	18.04
Python	>=3.6.9
Docker — ce	19.03.5
Docker API	1.40
nvidia-container-toolkit	>1.3.0-1
nvidia-container-runtime	3.4.0-1.
nvidia-docker2.	2.5.0-1.
NVIDIA 驱动程序	>455
Python-pip	>21.06
nvidia-pyindex	最新版本

## 用例详细信息

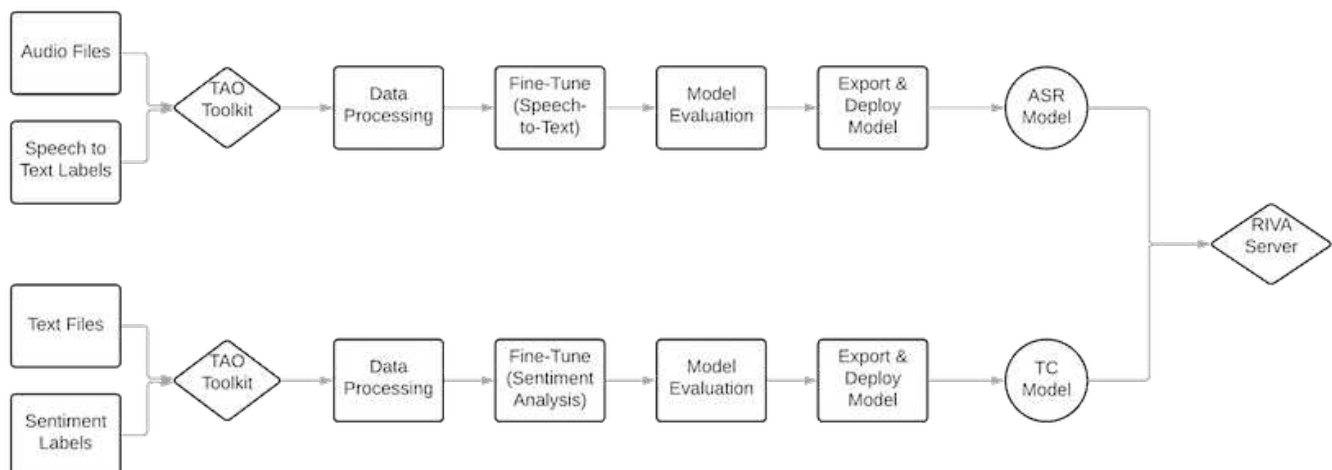
此解决方案适用场景的使用情形如下：

- 语音到文本
- 情感分析



语音到文本使用情形首先会为支持中心载入音频文件。然后，根据 Riva 所需的结构对该音频进行处理。如果尚未将音频文件拆分为其分析单元，则必须先执行此操作，然后再将音频传递给 Riva。处理完此音频文件后，它将成为 API 调用传递到 Riva 服务器。该服务器采用其托管的众多型号之一，并返回响应。此语音到文本（自动语音识别的一部分）将返回音频的文本表示形式。此时，管道将切换到情感分析部分。

对于情感分析，自动语音识别输出的文本将作为文本分类的输入。文本分类是用于将文本分类为任意数量的类别的 NVIDIA 组件。对于支持中心对话，情绪类别从正面到负面不等。可以使用一个组套来评估模型的性能，以确定微调步骤是否成功。



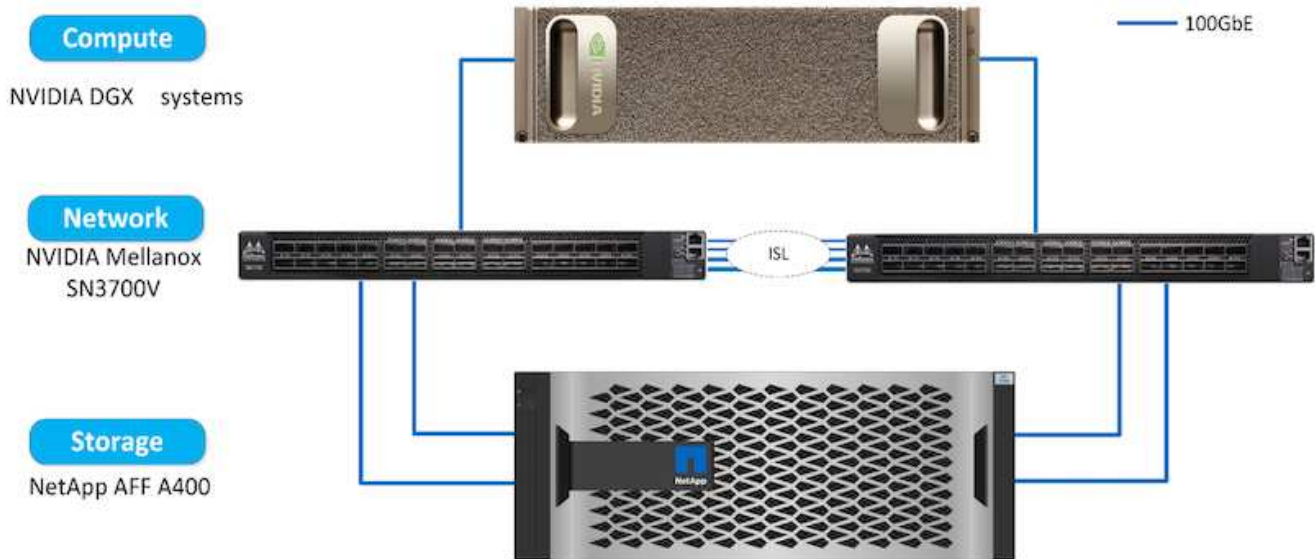
在 TAO 工具包中，语音到文本和情感分析也使用类似的管道。主要区别在于使用了微调模型所需的标签。TAO 工具包管道从处理数据文件开始。然后是经过预先培训的型号（来自 ["NVIDIA NGC 目录"](#)）。经过微调的模型会根据其相应的性能指标进行评估，如果性能比预先训练的模型更高，则会部署到 Riva 服务器。

## 设计注意事项

本节介绍此解决方案的不同组件的设计注意事项。

## 网络和计算设计

根据数据安全性的限制，所有数据都必须保留在客户的基础架构或安全环境中。



## 存储设计

NetApp DataOps 工具包是用于管理存储系统的主要服务。DataOps 工具包是一个 Python 库，可使开发人员，数据科学家，开发运营工程师和数据工程师轻松执行各种数据管理任务，例如近乎即时地配置新的数据卷或 JupyterLab 工作空间，近乎即时地克隆数据卷或 JupyterLab 工作空间，以及接近瞬时的数据卷快照或 JupyterLab 工作空间快照，以实现可追溯性或基线化。此 Python 库可以用作命令行实用程序或函数库，可以导入到任何 Python 程序或 Jupyter Notebook 中。

## RIVA 最佳实践

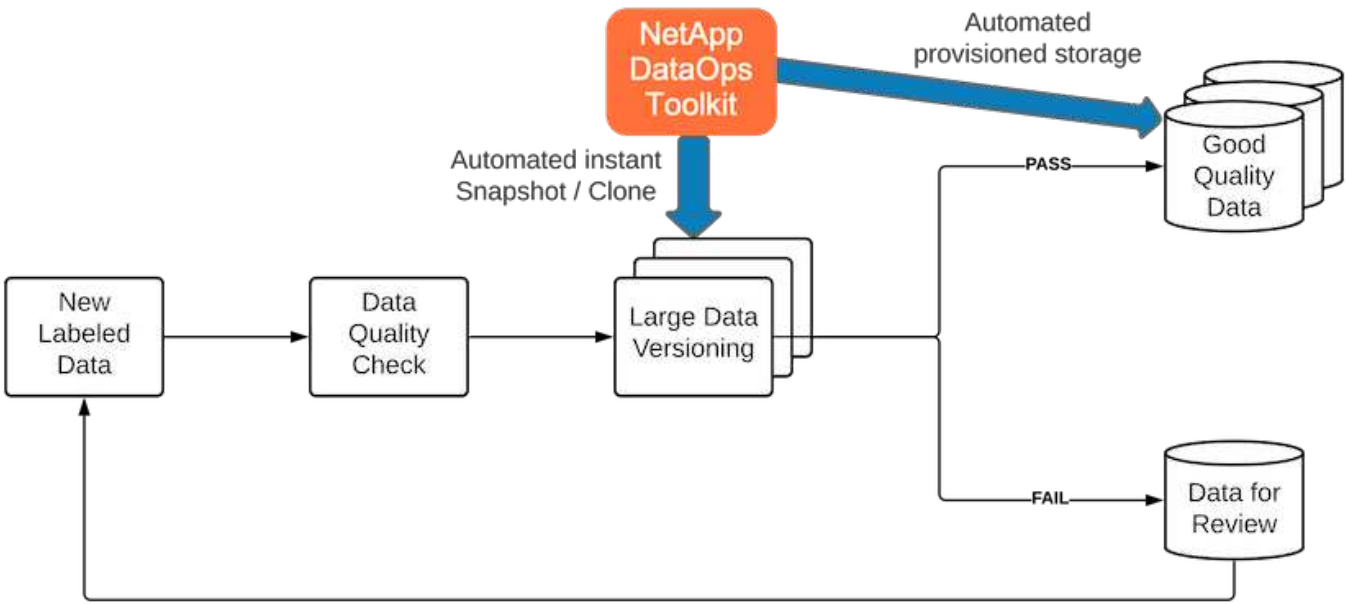
NVIDIA 提供了多种常规功能 "最佳数据实践" 要使用 Riva，请执行以下操作：

- \* 如果可能, 请使用无损音频格式。 \* 使用像 mp3 这样的有损编解码器可能会降低质量。
- \* 扩充训练数据。 \* 在音频训练数据中增加背景噪声最初会降低准确性, 但同时提高稳定性。
- \* 如果使用的是擦文本, 请限制词汇大小。 \* 许多在线源都包含拼写错误或辅助发音以及不常见的词。删除这些内容可以改进语言模式。
- \* 如果可能, 请使用最小采样速率 16kHz\*。但是, 请尝试不要重新采样, 因为这样做会降低音频质量。

除了这些最佳实践之外，客户还必须优先收集具有代表性的样本数据集，并为管道的每个步骤提供准确的标签。换言之，样本数据集应按比例反映目标数据集所示的指定特征。同样，数据集标注器也有责任平衡标记的准确性

和速度，以便最大限度地提高数据的质量和数量。例如，此支持中心解决方案需要音频文件，带标签的文本和情感标签。此解决方案的顺序性意味着从管道开始的错误会一直传播到管道的末尾如果音频文件质量较差，则文本抄录和情感标签也会是。

此错误传播方式与此类似，适用场景是对接受过此数据培训的模型进行的。如果情感预测 100% 准确，但语音到文本模式表现不佳，则最终管道将受初始音频到文本抄录的限制。开发人员必须单独考虑每个模型的性能，并将其作为更大管道的组成部分。在这种特定情况下，最终目标是开发一个能够准确预测情绪的渠道。因此，评估管道的总体指标是感受的准确性，而语音到文本的文字记录直接影响到这一点。



NetApp DataOps 工具包利用其近乎瞬时的数据克隆技术补充了数据质量检查管道。必须对每个标记的文件进行评估，并将其与现有标记的文件进行比较。在各种数据存储系统之间分布这些质量检查可确保快速高效地执行这些检查。

### 部署支持中心观点分析

部署解决方案涉及以下组件：

1. NetApp DataOps 工具包
2. NGC 配置
3. NVIDIA Riva 服务器
4. NVIDIA TAO 工具包
5. 将 TAO 模型导出到 Riva

要执行部署，请完成以下步骤：

#### NetApp DataOps 工具包：支持中心情感分析

以使用 "NetApp DataOps 工具包"，完成以下步骤：

1. 通过 Pip 安装工具包。

```
python3 -m pip install netapp-dataops-traditional
```

## 2. 配置数据管理

```
netapp_dataops_cli.py config
```

### NGC 配置：支持中心情感分析

进行设置 "NVIDIA NGC"，完成以下步骤：

#### 1. 下载 NGC。

```
wget -O ngccli_linux.zip  
https://ngc.nvidia.com/downloads/ngccli_linux.zip && unzip -o  
ngccli_linux.zip && chmod u+x ngc
```

#### 2. 将当前目录添加到路径。

```
echo "export PATH=\"\$PATH:$(pwd)\"" >> ~/.bash_profile && source  
~/.bash_profile
```

#### 3. 您必须配置 NGC 命令行界面以供使用，才能运行命令。出现提示时，输入以下命令，包括 API 密钥。

```
ngc config set
```

对于不基于 Linux 的操作系统，请访问 [此处](#)。

### NVIDIA Riva 服务器：支持中心情感分析

进行设置 "NVIDIA RIVA"，完成以下步骤：

#### 1. 从 NGC 下载 Riva 文件。

```
ngc registry resource download-version  
nvidia/riva/riva_quickstart:1.4.0-beta
```

2. 初始化 Riva 设置 (Riva\_init.sh)。
3. 启动 Riva 服务器 (Riva\_start.sh)。
4. 启动 Riva 客户端 (Riva\_start\_client.sh)。

5. 在 Riva 客户端中，安装音频处理库（"FFmpeg"）

```
apt-get install ffmpeg
```

6. 启动 "Jupyter" 服务器
7. 运行 Riva 推理管道笔记本电脑。

### NVIDIA TAO 工具包：支持中心情感分析

要设置 NVIDIA TAO 工具包，请完成以下步骤：

1. 准备并激活 "虚拟环境" 适用于 TAO 工具包。
2. 安装 "所需的软件包"。
3. 手动拉取训练和微调期间使用的图像。

```
docker pull nvcr.io/nvidia/tao/tao-toolkit-pyt:v3.21.08-py3
```

4. 启动 "Jupyter" 服务器
5. 运行 TAO 微调笔记本电脑。

### 将 TAO 模型导出到 Riva：支持中心情感分析

以使用 "Riva 中的 TAO 工具包模型"，完成以下步骤：

1. 将模型保存在 TAO 微调笔记本中。
2. 将经过 TAO 训练的型号复制到 Riva 型号目录。
3. 启动 Riva 服务器（Riva\_start.sh）。

### 部署障碍

在开发您自己的解决方案时，请谨记以下几点：

- 首先安装 NetApp DataOps 工具包，以确保数据存储系统以最佳状态运行。
- 必须先安装 NVIDIA NGC，再安装其他任何版本，因为它会对映像和型号的下载进行身份验证。
- 必须先安装 Riva，然后再安装 TAO 工具包。Riva 安装会根据需要配置 Docker 守护进程以提取映像。
- DGX 和 Docker 必须能够访问 Internet 才能下载这些型号。

### 验证结果

如上一节所述，只要有两个或更多机器学习模型按顺序运行，错误就会传播到整个管道中。对于此解决方案，句子的感受是衡量公司股票风险水平的最重要因素。语音到文本模式虽然对管道至关重要，但在预测情感之前，它是预处理单元。真正重要的是基本事实句子与预测的句子之间的情感差异。此选项用作错误率（WER）一词的代理。语音到文本

的准确性非常重要，但 WER 不会直接用于最终管道指标。

```
PIPELINE_SENTIMENT_METRIC = MEAN(DIFF(GT_sentiment, ASR_sentiment))
```

可以根据每句话的 F1 分数，回顾和精确度计算这些情感指标。然后，可以汇总结果并将其显示在混乱列表中，同时还可以显示每个指标的置信区间。

使用传输学习的优势在于，模型性能提高，只需满足极少的数据要求，培训时间和成本。此外，还应将经过微调的模型与其基线版本进行比较，以确保传输学习可以提高性能，而不会影响性能。换言之，经过微调的模型在支持中心数据上的性能应比预先训练的模型更好。

渠道评估

测试用例	详细信息
测试编号	渠道感受指标
测试前提条件	经过微调的语音到文本和情感分析模型模型
预期结果	经过微调的模型的情感指标比原始的预先训练模型的性能更好。

渠道感受指标

- 1. 计算基线模型的情感指标。
- 2. 计算经过微调的模型的情感指标。
- 3. 计算这些指标之间的差异。
- 4. 计算所有句子之间的差异平均值。

视频和演示

有两台笔记本电脑包含情感分析管道： [""Support-Center-Mode-Transfer-Learning-and-fine-Tuned.ipynb ""](#) 和 [""Support-Center-sentiment-Analysis-Pipeline.ipynb ""](#)。这些笔记本电脑共同演示了如何开发管道，以便使用根据用户数据微调的最先进深度学习模型，载入支持中心数据并从每句话中提取情感。

支持中心—情感分析管道 .ipynb

此笔记本电脑包含推理 Riva 管道，用于载入音频，将其转换为文本以及提取情感以供外部信息板使用。如果尚未下载和处理数据集，则会自动下载和处理该数据集。该笔记本电脑的第一部分是语音到文本，用于将音频文件转换为文本。接下来是 "情感分析" 部分，该部分将提取每个文本句子的感受，并以类似于建议的信息板的格式显示这些结果。



此笔记本电脑必须在模型训练和微调之前运行，因为必须下载并转换为正确格式的 mp3 数据集。



# Call Center - Sentiment Analysis Pipeline

This notebook demonstrates how to build a pipeline for sentiment analysis of call center conversations. The goal of this pipeline is to develop sentiment analysis for use within an external dashboard.

This tutorial will guide you through the use of [NVIDIA's RIVA](#) for automatic speech recognition and text classification. This tutorial uses NetApp cloud storage for data storage and a pre-trained RIVA model.

## Channels

These are the channels on which RIVA is hosting models.

- speech: 51051
- voice: 61051

These channels **must** be aligned with `riva_speech_api_port` and `riva_vision_api_port` within `config.sh`

```
In [4]: speech_channel = "localhost:51051"
voice_channel = "localhost:61051"
```

## Speech-To-Text

Automatic Speech Recognition (ASR) takes as input an audio stream or audio buffer and returns one or more text transcripts, along with additional optional metadata. ASR represents a full speech recognition pipeline that is GPU accelerated with optimized performance and accuracy. ASR supports synchronous and streaming recognition modes.

For more information on NVIDIA RIVA's Automatic Speech Recognition, visit [here](#).

## Constants

Use these constants to affect different aspects of this pipeline:

- `DATA_DIR` : base folder where data is stored
- `DATASET_NAME` : name of the call center dataset
- `COMPANY_DATE` : folder name identifying the particular call center conversation

支持中心—模型训练和微调。 **ipynb**

在执行此笔记本电脑之前，必须先设置 TAO 工具包虚拟环境（有关安装说明，请参见 "命令概述" 中的 "TAO 工具包" 一节）。

此笔记本电脑依靠 TAO 工具包根据客户数据微调深度学习模型。与上一个笔记本电脑一样，这一款笔记本电脑分为两个部分，分别用于语音到文本和情感分析组件。每个部分都将介绍数据处理，模型培训和微调，结果评估和模型导出。最后，我们将在最后一节中介绍如何部署这两种经过微调的型号，以便在 Riva 中使用。

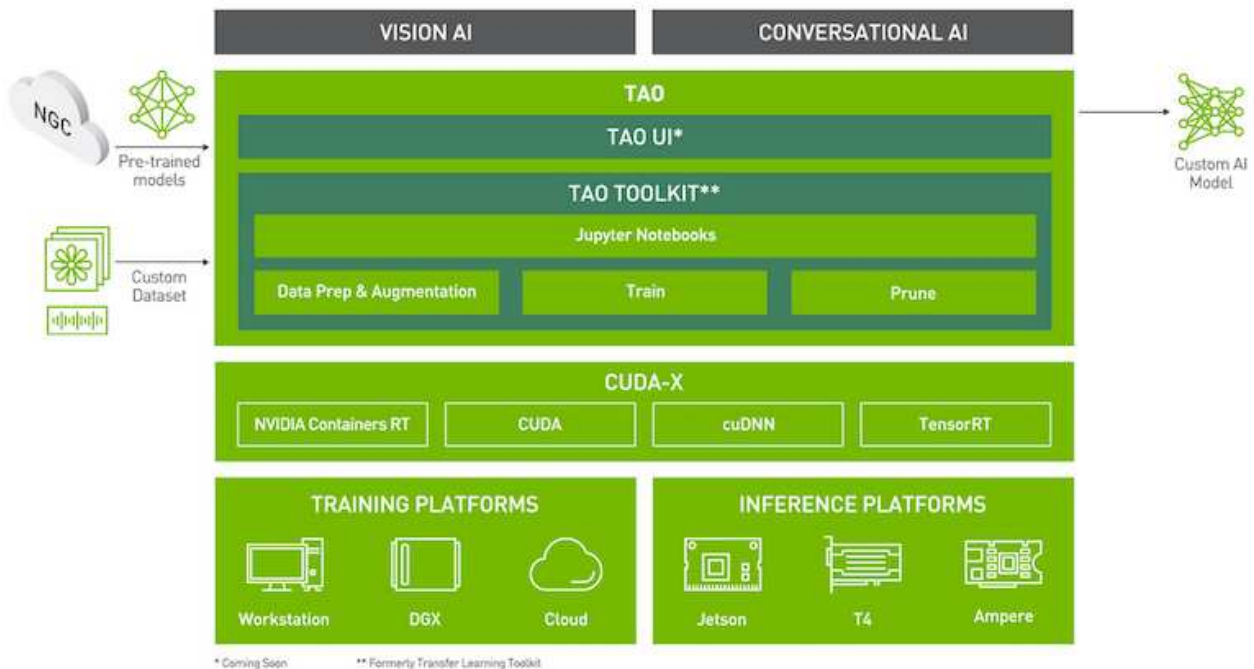


# Call Center - Model Transfer Learning and Fine-Tuning

TAO Toolkit is a python based AI toolkit for taking purpose-built pre-trained AI models and customizing them with your own data. Transfer learning extracts learned features from an existing neural network to a new one. Transfer learning is often used when creating a large training dataset is not feasible in order to enhance the base performance of state-of-the-art models.

For this call center solution, the speech-to-text and sentiment analysis models are fine-tuned on call center data to augment the model performance on business specific terminology.

For more information on the TAO Toolkit, please visit [here](#).



## Installing necessary dependencies

For ease of use, please install TAO Toolkit inside a python virtual environment. We recommend performing this step first and then launching the notebook from the virtual environment. Please refer to the README for these instructions.

## 结论

随着客户体验日益被视为关键的竞争战场，AI 增强型全球支持中心成为几乎每个行业的公司都无法忽视的关键组成部分。本技术报告中建议的解决方案已经证明能够为交付如此卓越的客户体验提供支持，现在的挑战是确保企业采取行动来实现 AI 基础架构和工作流的现代化。

AI 在客户服务中的最佳实施方式不是取代人工代理。相反，AI 可以通过实时情感分析，争议上报和多模式情感计算，让他们能够创造卓越的客户体验，从而检测语言，非语言和面部线索，通过这些线索，全面的 AI 模型可以大规模地提出建议，并补充个人代理可能缺少的内容。AI 还可以更好地匹配具有当前可用代理的特定客户。利用 AI，企业可以从客户对提供商产品，服务和品牌形象的想法和印象中汲取宝贵的意见。

解决方案还可用于为支持代理构建时间序列数据，以用作客观的性能评估指标。传统的客户满意度调查通常缺乏

足够的响应。通过收集员工和客户的长期情绪，雇主可以就支持代理的表现做出明智的决策。

NetApp，SFL 科学，开源业务流程框架和 NVIDIA 相结合，将最新技术作为托管服务集于一身，并具有极大的灵活性，可加快技术采用速度，加快新 AI/ML 应用程序的上市速度。这些高级服务在内部环境中提供，可轻松移植到云原生环境以及混合部署架构中。

## 从何处查找追加信息

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：

- 3D 交互式演示

["www.netapp.com/ai"](http://www.netapp.com/ai)

- 直接与 NetApp AI 专家联系

["https://www.netapp.com/artificial-intelligence/"](https://www.netapp.com/artificial-intelligence/)

- 《采用 NetApp 解决方案的 NVIDIA 基本命令平台》简介

<https://www.netapp.com/pdf.html?item=/media/32792-DS-4145-NVIDIA-Base-Command-Platform-with-NetApp.pdf>

- NetApp 的 AI 10 大理由信息图

["https://www.netapp.com/us/media/netapp-ai-10-good-reasons.pdf"](https://www.netapp.com/us/media/netapp-ai-10-good-reasons.pdf)

- 医疗保健领域的人工智能：深入学习如何在 lung-CT 扫描中识别 COVID-19 病变白皮书

<https://www.netapp.com/pdf.html?item=/media/31240-WP-7342.pdf>

- 医疗保健领域的人工智能：监控医疗保健环境中的面部掩模使用情况白皮书

<https://www.netapp.com/pdf.html?item=/media/37490-NA-611-Monitoring-face-mask-usage-in-healthcare-settings.pdf>

- 医疗保健领域的人工智能：诊断成像技术报告

<https://www.netapp.com/pdf.html?item=/media/7395-tr4811.pdf>

- AI for Retail：使用 NVIDIA Riva 的 NetApp 对话 AI

["内容提要"](#)

- NetApp ONTAP AI 解决方案简介

<https://www.netapp.com/pdf.html?item=/media/6736-sb-3939.pdf>

- NetApp DataOps 工具包解决方案简介

<https://www.netapp.com/pdf.html?item=/media/21480-SB-4111-1220-NA-Data-Science-Toolkit.pdf>

- NetApp AI 控制平面解决方案简介

<https://www.netapp.com/pdf.html?item=/media/6737-sb-4055.pdf>

- 《利用 Data Drive AI 转变行业》电子书

["https://www.netapp.com/us/media/na-337.pdf"](https://www.netapp.com/us/media/na-337.pdf)

- NetApp EF 系列 AI 解决方案简介

<https://www.netapp.com/pdf.html?item=/media/26708-SB-4136-NetApp-AI-E-Series.pdf>

- NetApp AI 和联想 ThinkSystem for AI 推理解决方案简介

<https://www.netapp.com/pdf.html?item=/media/25316-SB-4129.pdf>

- NetApp AI 和联想 ThinkSystem for Enterprise AI 和 ML 解决方案简介

<https://www.netapp.com/pdf.html?item=/media/25317-SB-4128.pdf>

- NetApp 和 NVIDIA —重新定义人工智能视频的可能

<https://www.youtube.com/watch?v=38xw65SteUc>

## Azure 中的分布式培训—点击率预测

### TR-4904： Azure 中的分布式培训—点击率预测

Rick Huang， Verron Martina， Muncher， NetApp

数据科学家的工作重点应放在机器学习（ML）和人工智能（AI）模型的培训和调整上。但是，根据 Google 的研究，数据科学家花费大约 80% 的时间来研究如何使其模型能够与企业应用程序结合使用并大规模运行。

要管理端到端 AI/ML 项目，需要更广泛地了解企业组件。虽然 DevOps 已接管了这些类型的组件的定义，集成和部署，但 ML 操作的目标是类似的流程，其中包括 AI/ML 项目。要了解端到端 AI/ML 管道在企业中涉及的内容，请参见以下所需组件列表：

- 存储
- 网络
- 数据库
- 文件系统
- 容器
- 持续集成和持续部署（CI/CD）管道
- 集成开发环境（IDE）
- 安全性
- 数据访问策略
- 硬件

- 云
- 虚拟化
- 数据科学工具集和库

## 目标受众

数据科学领域涉及 IT 和业务领域的多个领域：

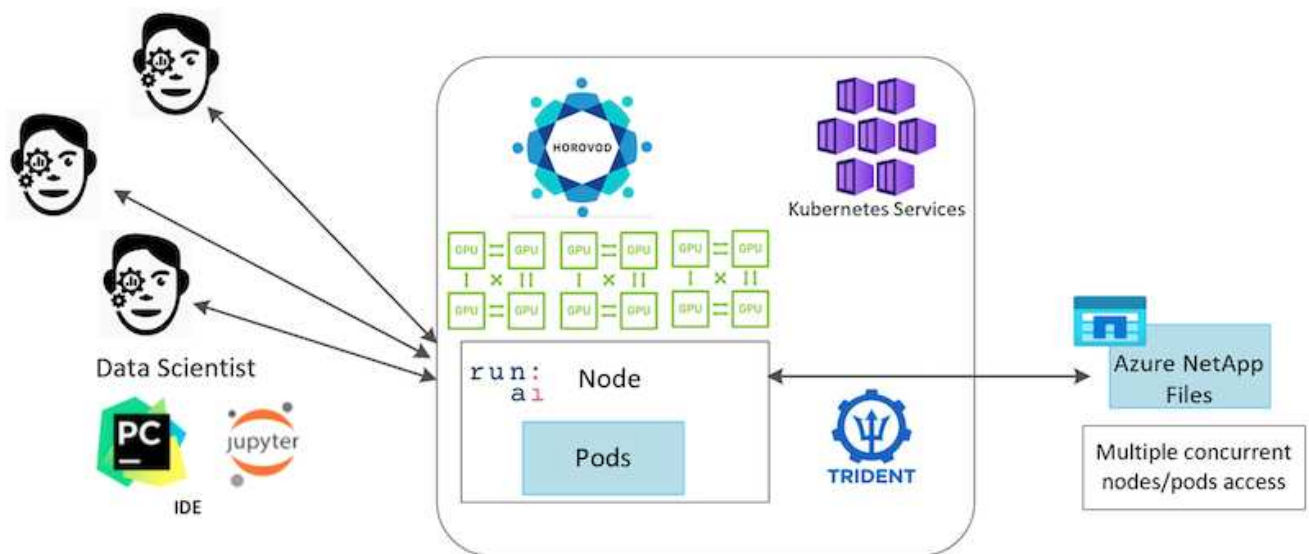
- 数据科学家需要灵活地使用自己选择的工具和库。
- 数据工程师需要了解数据的流动方式及其所在位置。
- DevOps 工程师需要使用工具将新的 AI/ML 应用程序集成到其 CI/CD 管道中。
- 云管理员和架构师需要能够设置和管理 Azure 资源。
- 业务用户希望能够访问 AI/ML 应用程序。

在本技术报告中，我们将介绍 Azure NetApp Files，快速 AI，dask 和 Azure 如何帮助这些角色为业务带来价值。

## 解决方案概述

此解决方案遵循 AI/ML 应用程序的生命周期。我们从数据科学家的工作入手，确定准备数据和训练模型所需的不同步骤。通过利用 dask 上的快速访问功能，我们可以在 Azure Kubernetes Service（AKS）集群中执行分布式培训，与传统的 Python 科学学习方法相比，可以大幅缩短培训时间。为了完成整个周期，我们将管道与 Azure NetApp Files 集成在一起。

Azure NetApp Files 提供各种性能层。客户可以从标准层开始，无需移动任何数据即可无中断地横向扩展和纵向扩展到高性能层。数据科学家可以利用此功能大规模训练模型，而不会出现任何性能问题，从而避免集群中出现任何数据孤岛，如下图所示。



## 技术概述

此页面概述了此解决方案中使用的技术。

## Microsoft 和 NetApp

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。

### Azure NetApp Files

Azure NetApp Files 服务是一种企业级高性能计量文件存储服务。Azure NetApp Files 支持任何工作负载类型，默认情况下具有高可用性。您可以选择服务和性能级别，并通过服务设置 Snapshot 副本。Azure NetApp Files 是一项 Azure 第一方服务，用于在云中迁移和运行要求最严苛的企业文件工作负载，包括数据库，SAP 和高性能计算应用程序，并且不会更改代码。

此参考架构为 IT 组织提供了以下优势：

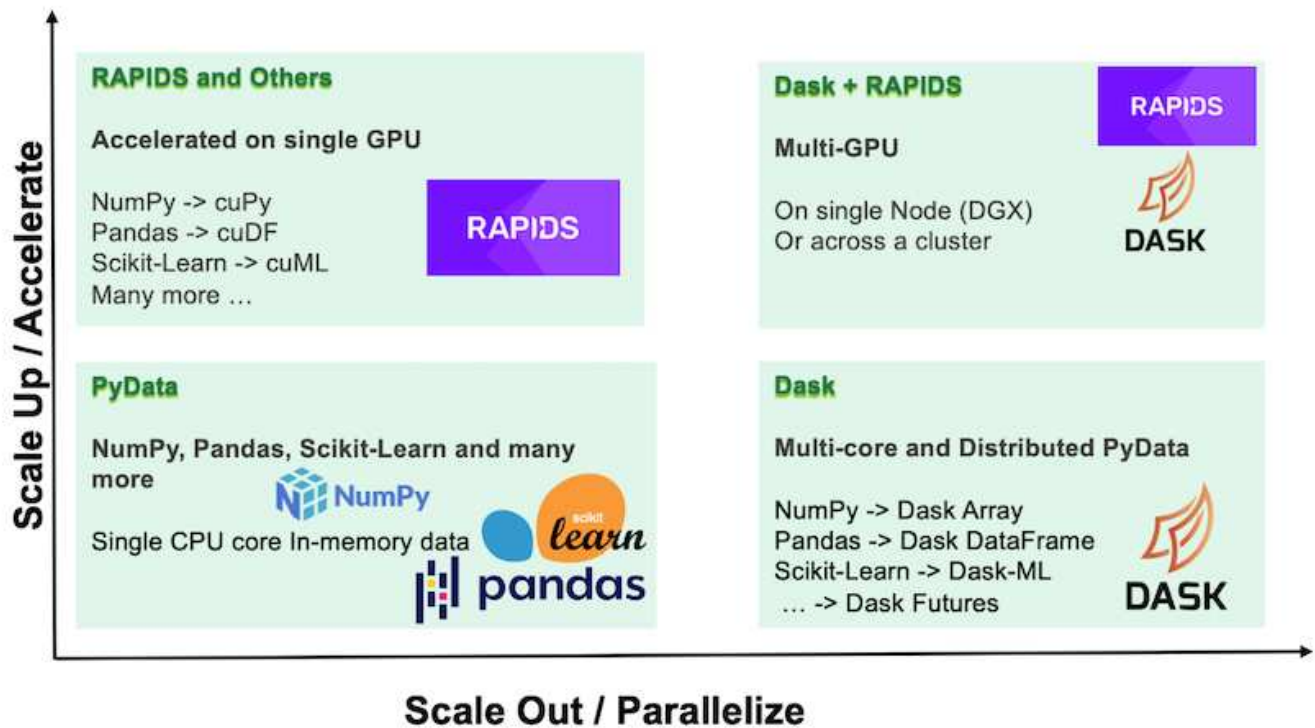
- 消除设计复杂性
- 支持独立扩展计算和存储
- 支持客户从小规模入手，无缝扩展
- 为各种性能和成本点提供一系列存储层

### Dask 和 NVIDIA 快速流概述

Dask 是一款开源并行计算工具，可在多台计算机上扩展 Python 库，并加快处理大量数据的速度。它提供的 API 类似于单线程传统 Python 库，例如熊猫，Numpy 和 sciKit 学习。因此，原生 Python 用户不必对其现有代码进行大量更改，即可在整个集群中使用资源。

NVIDIA RAID 是一套开源库，可以完全在 GPU 上运行端到端 ML 和数据分析工作流。通过与 dask 结合使用，您可以轻松地从 GPU 工作站（纵向扩展）扩展到多节点，多 GPU 集群（横向扩展）。

要在集群上部署 dask，您可以使用 Kubernetes 进行资源编排。您还可以根据进程要求纵向或横向扩展工作节点，进而有助于优化集群资源消耗，如下图所示。



## 软件要求

下表列出了此解决方案所需的软件要求。

软件	version
Azure Kubernetes Service	1.18.14
流和 dask 容器映像	存储库: "rapidsai/rapidsai" 标记: 0.17-cuda11.0-runtime-ubuntu18.04
NetApp Trident	20.01.1
掌舵	3.0.0

## 云资源要求

此页面介绍了 Azure NetApp Files 云资源的配置。

### 配置 Azure NetApp Files

按照中所述配置 Azure NetApp Files ["快速入门：设置 Azure NetApp Files 并创建 NFS 卷"](#)。

您可以继续阅读 "为 Azure NetApp Files 创建 NFS 卷" 一节，因为您要通过 Trident 创建卷。在继续之前，请完成以下步骤：

1. 注册 Azure NetApp Files 和 NetApp 资源提供商（通过 Azure Shell）（["链接。"](#)）。
2. 在 Azure NetApp Files 中创建帐户（["链接。"](#)）。
3. 设置容量池（至少 4 TB 标准版或高级版，具体取决于您的需要）（["链接。"](#)）。下表列出了在云中设置的网络配置要求。dask 集群和 Azure NetApp Files 必须位于同一个 Azure 虚拟网络（vNet）或对等 vNet



中。

Resources	类型 / 版本
Azure Kubernetes Service	1.18.14
代理节点	3 个 Standard_DS2_v2
GPU 节点	3 个 Standard_NC6s_v3
Azure NetApp Files	标准容量池
容量（以 TB 为单位）	4.

## 点击率预测用例摘要

此用例基于公开发布的 **"TB 单击 Logs"** 数据集来自 **"Criteo AI 实验室"**。随着 ML 平台和应用程序的最新发展，我们现在非常关注大规模学习。点击率（CTR）是指每 100 次在线广告曝光的平均点击次数（以百分比表示）。它已广泛用作各种行业垂直市场和用例的关键指标，包括数字营销，零售，电子商务和服务提供商。使用 CTR 作为潜在客户流量的重要指标的示例包括：

- \* 数字营销：\* 在中 **"Google 分析"**，可以使用 CTR 来衡量广告商或商家的关键字，广告和免费列表的表现。如果 CTR 较高，则表明用户发现您的广告和列表很有用且相关。CTR 还有助于实现关键字的预期 CTR，它是的组成部分 **"AD 排名"**。
- \* 电子商务：\* 除利用外 **"Google 分析"**，电子商务后端至少有一些访客统计信息。虽然这些统计信息乍看起来可能并不有用，但通常很容易阅读，并且可能比其他信息更准确。由此类统计信息组成的第一方数据集属于专有数据集，因此与电子商务销售商，买家和平台最相关。这些数据集可用于设置基准，并通过构建时间序列来与去年和昨天的结果进行比较，以供进一步分析。
- \* 零售：\* Brick-and-mortar 零售商可以将访客数量和客户数量与 CTR 相关联。客户数量可从其销售点历史记录中查看。零售商网站或广告流量的 CTR 可能会导致上述销售。忠诚度计划是另一个用例，因为从在线广告或其他网站转出的客户可能会加入该计划以获得奖励。零售商可以通过忠诚度计划赢得客户，并记录销售历史记录中的行为，从而构建一个建议系统，该系统不仅可以预测不同类别的消费者购买行为，还可以对优惠券进行个性化设置并减少流失。
- \* 服务提供商：\* 电信公司和互联网服务提供商拥有大量第一方用户遥测数据，可用于富有洞察力的 AI，ML 和分析用例。例如，电信可以利用其移动用户每天的 Web 浏览顶级域历史日志来微调现有模型，以生成最新的受众细分，预测客户行为，并与广告商协作发布实时广告，从而获得更好的在线体验。在这种数据驱动型营销工作流程中，CTR 是反映转换的一个重要指标。

在数字化营销背景下，**"Trigeo Terabyte 单击 Logs"** 现在是评估 ML 平台和算法可扩展性的参考数据集。通过预测点击率，广告宣传者可以选择最有可能对广告做出响应的访客，分析浏览历史记录，并根据用户兴趣显示最相关的广告。

本技术报告中提供的解决方案重点介绍了以下优势：

- Azure NetApp Files 在分布式或大规模培训方面的优势
- 支持 CUDA 的数据处理（cuDF，cuPy 等）和 ML 算法（cuML）
- 分布式培训的 dask 并行计算框架

基于快速 AI 和 Azure NetApp Files 构建的端到端工作流程展示了随机林模型训练时间大幅缩短两个数量级。与传统的熊猫方法相比，每天处理实际点击日志时，这种改进非常重要，其中每天包含 45 GB 的结构化表格数据（

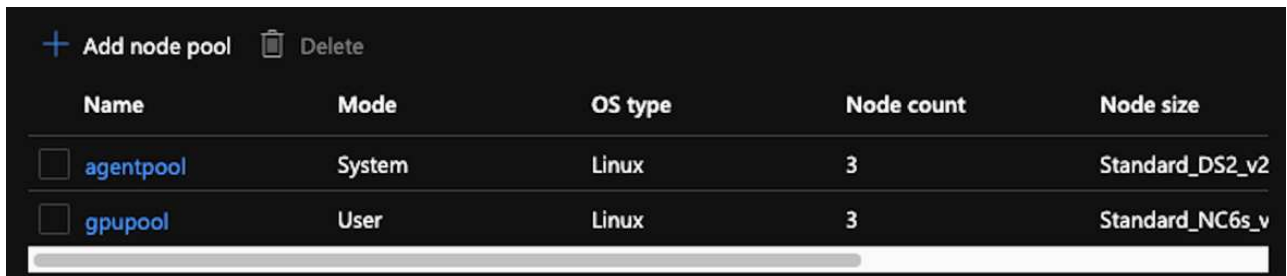
平均)。这相当于一个包含大约 200 亿行的 DataFrame。我们将在本技术报告中演示集群环境设置，框架和库安装，数据加载和处理，传统培训与分布式培训，可视化和监控，并比较关键的端到端运行时结果。

## 设置

### 安装和设置 AKS 集群

要安装和设置 AKS 集群，请参见网页 ["创建 AKS 集群"](#) 然后完成以下步骤：

1. 选择节点类型（系统 CPU 或辅助 GPU 节点）时，请选择以下选项：
  - a. 主系统节点应为标准 DS2v2（agentpool 默认三个节点）。
  - b. 然后，为名为 gpupool 的用户组（对于 GPU 节点）添加工作节点 Standard\_Nc6s\_v3 Pool（至少三个节点）。



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

2. 部署需要 5 到 10 分钟。完成后，单击 Connect to Cluster。
3. 要连接到新创建的 AKS 集群，请从本地环境（笔记本电脑 /PC）安装以下内容：
  - a. 使用的 Kubernetes 命令行工具 ["针对您的特定操作系统提供的说明"](#)
  - b. 文档中所述的 Azure 命令行界面，["安装 Azure 命令行界面"](#)
4. 要从终端访问 AKS 集群，请输入 az login 并输入凭据。
5. 运行以下两个命令：

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

6. 输入 Azure CLI：`kubectl get nodes`。
7. 如果所有六个节点均已启动且正在运行，如以下示例所示，则 AKS 集群已准备就绪并连接到本地环境

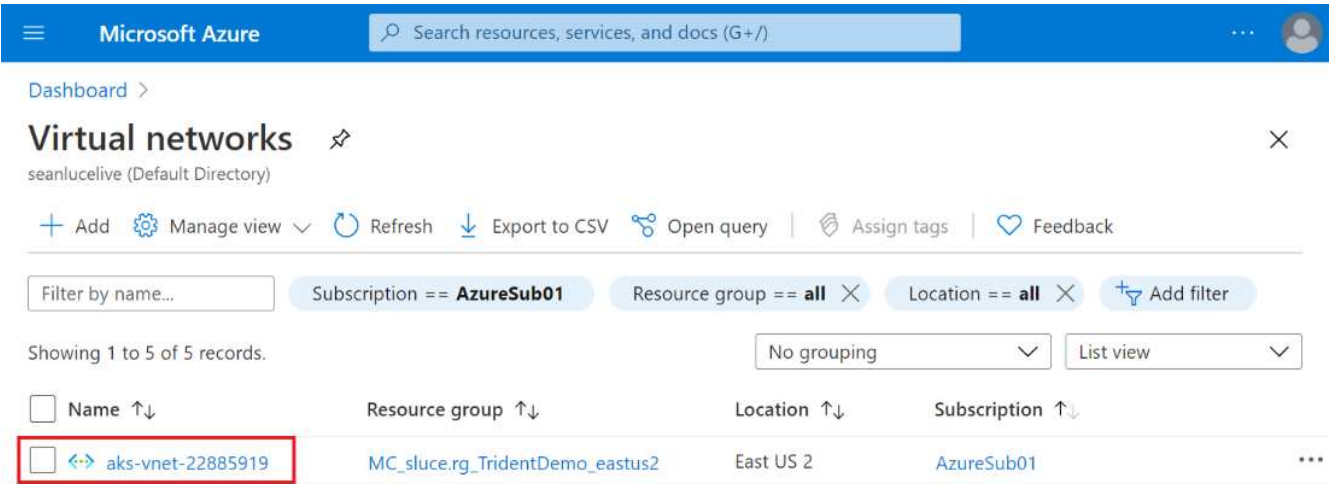
```
verronmartina@verron-mac-0 ~ % kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
aks-agentpool-34613062-vmss000000  Ready    agent    22m   v1.18.14
aks-agentpool-34613062-vmss000001  Ready    agent    22m   v1.18.14
aks-agentpool-34613062-vmss000002  Ready  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



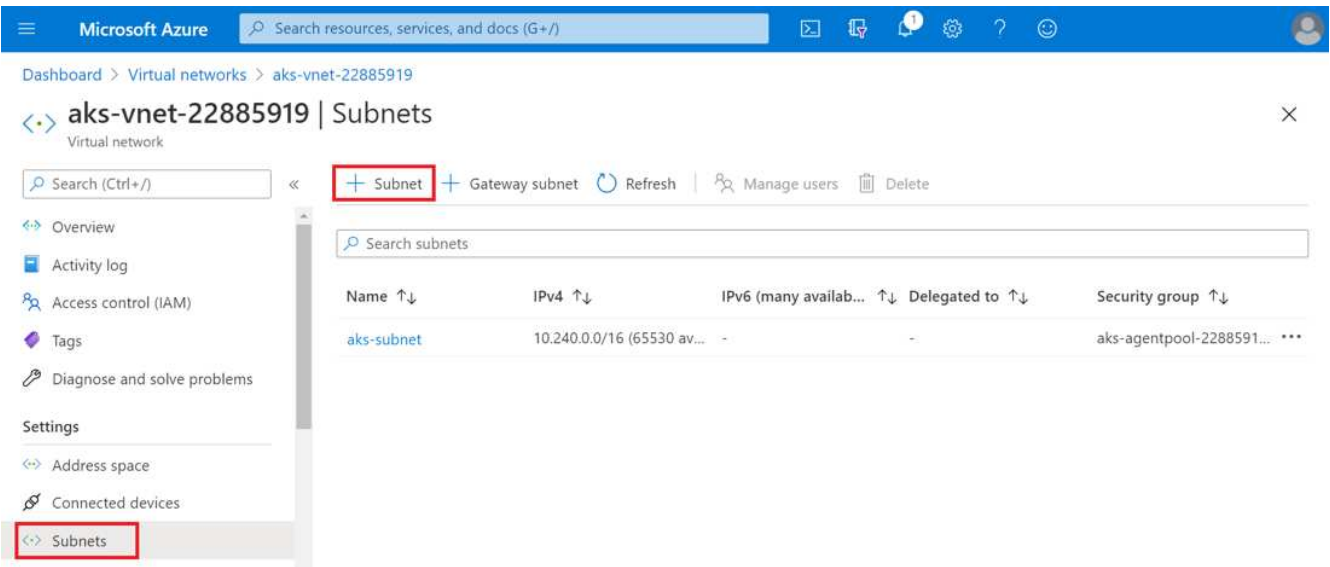
为 Azure NetApp Files 创建委派子网

要为 Azure NetApp Files 创建委派子网，请完成以下步骤：

- 1. 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 AK -vnet 等前缀。
- 2. 单击 vNet 的名称。



- 3. 单击子网，然后单击顶部工具栏中的 +Subnet 。



- 4. 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 \*

ANF.sn



Subnet address range \* ⓘ

10.0.0.0/24

10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

☐

Add IPv6 address space ⓘ

NAT gateway ⓘ

None



Network security group

None



Route table

None



### SERVICE ENDPOINTS

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

Services ⓘ

0 selected



### SUBNET DELEGATION

Delegate subnet to a service ⓘ

Microsoft.Netapp/volumes



OK

Cancel

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，通过此过程，您可以灵活地将其映射到不同的服务，例如 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。在本技术报告讨论 NFS 时， Azure NetApp Files 的主要优势包括：

- 为用户提供使用 Snapshot 副本的功能。
- 允许用户在 Azure NetApp Files 卷上存储大量数据。
- 在大型文件集上运行 Azure NetApp Files 卷的型号时，可以利用这些卷的性能优势。

## 对等 AKS vNet 和 Azure NetApp Files vNet

要将 AKS vNet 与 Azure NetApp Files vNet 建立对等关系，请完成以下步骤：

1. 在搜索字段中输入虚拟网络。
2. 选择 `vnet AK -vnet-name` 。单击它并在搜索字段中输入 "Peels" 。
3. 单击 +Add 。
4. 输入以下描述符：
  - a. 对等链路名称是 `aps-vnet-name_to_anf` 。
  - b. `subscriptionID` 和 Azure NetApp Files vNet 作为 vNet 对等配对节点。
  - c. 保留所有非星号部分的默认值。
5. 单击添加。

有关详细信息，请参见 ["创建，更改或删除虚拟网络对等关系"](#)。

## 安装 Trident

要使用 Helm 安装 Trident ，请完成以下步骤：

1. 安装 Helm （有关安装说明，请访问 ["source"](#)）。
2. 下载并解压缩 Trident 20.01.1 安装程序。

```
$wget  
$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`` 。

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

5. 使用 Helm （["source"](#)）：

- 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. 检查 Trident Pod 的状态。

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

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。

6. 为 AKS 设置 Azure NetApp Files 后端和存储类。

a. 创建 Azure 服务原则。

服务主体是 Trident 如何与 Azure 通信以操作 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": "",
  "password": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxx"
}
```

7. 创建 Trident 后端 json 文件，示例名称 anf-backend.json。

8. 使用首选文本编辑器，完成 anf-backend.json 文件中的以下字段：

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "fakec765-4774-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"
  }
}
```

9. 替换以下字段：

- `ssubscriptionID`。您的 Azure 订阅 ID
- 租户 ID。上一步中 `az ad sp` 输出中的 Azure 租户 ID。
- 客户端 ID。上一步中 `az ad sp` 输出中的 `appId`。
- `clientSecret`。上一步中 `az ad sp` 输出中的密码。

10. 指示 Trident 在 `trident` 命名空间中使用 `anf-backend.json` 作为配置文件创建 Azure NetApp Files 后端：

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

NAME	STORAGE DRIVER	UUID	STATE	VOLUMES
azurenetaappfiles_86181	azure-netapp-files	2ca85462-59ac-4946-be05-c03f5575a2ad	online	0

11. 创建存储类。Kubernetes 用户使用按名称指定存储类的 PVC 配置卷。指示 K8s 创建一个 `azurenetaappfiles`，该存储类引用上一步创建的 Trident 后端。
12. 为存储类和副本创建 YAML (`anf-storage-class.yaml`) 文件。

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: azurenetappfiles
provisioner: netapp.io/trident
parameters:
  backendType: "azure-netapp-files"
$kubectl create -f anf-storage-class.yaml

```

13. 验证是否已创建存储类。

```
kubectl get sc azurenetappfiles
```

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

使用 **Helm** 在 **AKS** 上设置带快速部署的 **dask**

要使用 **Helm** 在 **AKS** 上设置带快速部署的 **dask**，请完成以下步骤：

1. 创建一个命名空间以安装带有快速流的 **dask**。

```
kubectl create namespace rapids-dask
```

2. 创建一个 **PVC** 以存储点击率数据集：

a. 将以下 **YAML** 内容保存到文件中以创建 **PVC**。

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-criteo-data
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1000Gi
  storageClassName: azurenetappfiles

```

b. 将 **YAML** 文件应用于 **Kubernetes** 集群。

```
kubectl -n rapids-dask apply -f <your yaml file>
```

3. 克隆 apidsai git repository (["https://github.com/rapidsai/helm-chart"](https://github.com/rapidsai/helm-chart))。

```
git clone https://github.com/rapidsai/helm-chart helm-chart
```

4. 修改 values.yaml 并包括先前为员工和 Jupyter 工作空间创建的 PVC。

- a. 转至存储库的 rapidsai 目录。

```
cd helm-chart/rapidsai
```

- b. 更新 values.yaml 文件并使用 PVC 挂载卷。

```
dask:
  ...
  worker:
    name: worker
    ...
  mounts:
    volumes:
      - name: data
        persistentVolumeClaim:
          claimName: pvc-criteo-data
    volumeMounts:
      - name: data
        mountPath: /data
    ...
  jupyter:
    name: jupyter
    ...
  mounts:
    volumes:
      - name: data
        persistentVolumeClaim:
          claimName: pvc-criteo-data
    volumeMounts:
      - name: data
        mountPath: /data
    ...
```

5. 转到存储库的主目录，然后使用 Helm 在 AKS 上为三个辅助节点部署 dask。

```
cd ..
helm dep update rapidsai
helm install rapids-dask --namespace rapids-dask rapidsai
```

## Azure NetApp Files 性能层

您可以通过将现有卷移动到使用所需卷服务级别的另一个容量池来更改此卷的服务级别。借助此解决方案，客户可以从标准层中的小型数据集和少量 GPU 入手，并随着数据量和 GPU 的增加而横向扩展或纵向扩展到高级层。高级层提供的每 TB 吞吐量是标准层的四倍，并且可以执行纵向扩展，而无需移动任何数据即可更改卷的服务级别。

动态更改卷的服务级别

要动态更改卷的服务级别，请完成以下步骤：

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:/for	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。





3. 单击确定。

自动执行性能层更改

以下选项可用于自动执行性能层更改：

- 动态服务级别更改目前仍在公有预览中，默认情况下不会启用。要在 Azure 订阅上启用此功能，请参见本文档了解如何操作 ["动态更改卷的服务级别"](#)。
- 中提供了 Azure CLI 卷池更改命令 ["卷池更改文档"](#) 在以下示例中：

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

- PowerShell：["Set-AzNetAppFilesVolumePool cmdlet"](#) 更改 Azure NetApp Files 卷的池，如以下示例所示：

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

## 单击查看速率预测数据处理和模型训练

用于数据处理和模型培训的库

下表列出了用于构建此任务的库和框架。所有这些组件均已与 Azure 基于角色的访问和安全控制完全集成。

库 / 框架	Description
dask cuML	要使 ML 在 GPU 上运行，请使用 <a href="#">"cuML 库"</a> 可通过 dask 访问快速通道 cML 软件包。与基于 CPU 的方法相比，强强联合可以通过基于 GPU 的高性能实施实施常见的 ML 算法，包括集群，维度缩减和回归方法，提供高达 100 倍的速度。
dask cudf	CUDF 包括支持 GPU 加速提取，转换，加载（ETL）的各种其他功能，例如数据子设置，转换，单热编码等。快速发展团队会维护 <a href="#">"dask-cudf 库"</a> 其中包括使用 dask 和 cuDF 的帮助程序方法。
Scikit 学习	Scikit Learning 提供了数十种内置机器学习算法和模型，称为评估器。每个 <a href="#">"估算器"</a> 可以使用将其安装到某些数据中 <a href="#">"适合"</a> 方法

我们使用两台笔记本电脑构建 ML 管道进行比较；一台是传统的熊猫科学学习方法，另一台是使用快速和快速的分布式培训。每台笔记本电脑均可单独进行测试，以查看时间和规模方面的性能。我们会分别介绍每台笔记本电脑，以展示使用快速流和 dask 进行分布式培训的优势。

在熊猫中加载 **Criteo** 单击 **Logs Day 15**，然后训练一个 **sc**科学 学习随机林模型

本节介绍如何使用熊猫和 dask DataFrames 从 Criteo TB 数据集中加载 Click Logs 数据。在广告交换的数字广告中，使用情形与此相关，它可以预测是否会点击广告，或者如果交换在自动管道中使用的模型不准确，从而构建用户的个人资料。

我们从 Click Logs 数据集加载了第 15 天的数据，总计 45 GB。在 Jupyter 笔记本电脑中运行以下单元 `CT-PandasRF-colled.ipynb` 创建一个包含前 5，000 万行的熊猫 DataFrame，并生成一个 scide-Learn 随机林模型。

```

%%time
import pandas as pd
import numpy as np
header = ['col'+str(i) for i in range (1,41)] #note that according to
criteo, the first column in the dataset is Click Through (CT). Consist of
40 columns
first_row_taken = 50_000_000 # use this in pd.read_csv() if your compute
resource is limited.
# total number of rows in day15 is 20B
# take 50M rows
"""
Read data & display the following metrics:
1. Total number of rows per day
2. df loading time in the cluster
3. Train a random forest model
"""
df = pd.read_csv(file, nrows=first_row_taken, delimiter='\t',
names=header)
# take numerical columns
df_sliced = df.iloc[:, 0:14]
# split data into training and Y
Y = df_sliced.pop('col1') # first column is binary (click or not)
# change df_sliced data types & fillna
df_sliced = df_sliced.astype(np.float32).fillna(0)
from sklearn.ensemble import RandomForestClassifier
# Random Forest building parameters
# n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
rf_model = RandomForestClassifier(max_depth=max_depth,
n_estimators=n_trees)
rf_model.fit(df_sliced, Y)

```

要使用经过培训的随机林模型执行预测，请在此笔记本电脑中运行以下段落。为了避免重复，我们采用了自第 15 天起的最后 100 万行作为测试集。该单元格还会计算预测准确性，其定义为模型准确预测用户是否单击 AD。要查看此笔记本中任何不熟悉的组件，请参见 ["官方科学知识工具包学习文档"](#)。

```
# testing data, last 1M rows in day15
test_file = '/data/day_15_test'
with open(test_file) as g:
    print(g.readline())

# dataframe processing for test data
test_df = pd.read_csv(test_file, delimiter='\t', names=header)
test_df_sliced = test_df.iloc[:, 0:14]
test_Y = test_df_sliced.pop('coll')
test_df_sliced = test_df_sliced.astype(np.float32).fillna(0)
# prediction & calculating error
pred_df = rf_model.predict(test_df_sliced)
from sklearn import metrics
# Model Accuracy
print("Accuracy:", metrics.accuracy_score(test_Y, pred_df))
```

在 **dask** 中加载第 15 天，训练一个 **dask cuML** 随机林模型

按照与上一节类似的方式，在熊猫中加载 Criteo 单击 Logs Day 15，然后训练一个 cscit-Learn 随机林模型。在此示例中，我们使用 dask cuDF 执行了 DataFrame 加载，并在 dask cuML 中训练了一个随机林模型。我们比较了本节中培训时间和规模的差异 ["" 培训时间比较。""](#)

#### Criteo\_dask\_rf.ipynb

此笔记本电脑将导入 NumPy，累积和必要的 dask 库，如以下示例所示：

```
import cuml
from dask.distributed import Client, progress, wait
import dask_cudf
import numpy as np
import cudf
from cuml.dask.ensemble import RandomForestClassifier as cumlDaskRF
from cuml.dask.common import utils as dask_utils
```

启动 dask 客户端 ()。

```
client = Client()
```

如果集群配置正确，您可以查看工作节点的状态。

```
client
workers = client.has_what().keys()
n_workers = len(workers)
n_streams = 8 # Performance optimization
```

在我们的 AKS 集群中，将显示以下状态：

Client	Cluster
<b>Scheduler:</b> tcp://rapidsai-scheduler:8786	<b>Workers:</b> 3
<b>Dashboard:</b> <a href="/proxy/rapidsai-scheduler:8787/status">/proxy/rapidsai-scheduler:8787/status</a>	<b>Cores:</b> 3
	<b>Memory:</b> 354.55 GB

请注意，dask 采用了延迟执行模式：dask 不是即时执行处理代码，而是构建定向的环状图（DAG）执行。DAG 包含一组任务及其互动，每位员工都需要执行这些任务。此布局意味着，只有在用户指示 dask 以某种方式执行任务后，这些任务才会运行。使用 dask，您有三个主要选项：

- 在 DataFrame 上调用 `compute()`。\* 此调用将处理所有分区，然后将结果返回给计划程序，以便最终聚合和转换为 cuDF DataFrame。除非计划程序节点内存不足，否则应谨慎使用此选项，并且仅在结果大幅减少时使用。
- \* 在 DataFrame 上调用 `persiste()`。\* 此调用执行图形，但它不会将结果返回到计划程序节点，而是在整个集群的内存中保留这些结果，以便用户可以在管道中重复使用这些中间结果，而无需重新运行相同的处理。
- \* 在 DataFrame 上调用 `head()`。\* 与 cuDF 一样，此调用会将 10 条记录返回到计划程序节点。此选项可用于快速检查 DataFrame 是否包含所需的输出格式，或者记录本身是否合理，具体取决于您的处理和计算结果。

因此，除非用户调用其中任一操作，否则员工将处于闲置状态，等待计划程序启动处理。这种延迟执行模式在现代并行和分布式计算框架（如 Apache Spark）中很常见。

下一段使用 dask cuML 进行分布式 GPU 加速计算，以此训练随机林模型，并计算模型预测准确性。

```

Adsf
# Random Forest building parameters
n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
cuml_model = cumlDaskRF(max_depth=max_depth, n_estimators=n_trees,
n_bins=n_bins, n_streams=n_streams, verbose=True, client=client)
cuml_model.fit(gdf_sliced_small, Y)
# Model prediction
pred_df = cuml_model.predict(gdf_test)
# calculate accuracy
cu_score = cuml.metrics.accuracy_score( test_y, pred_df )

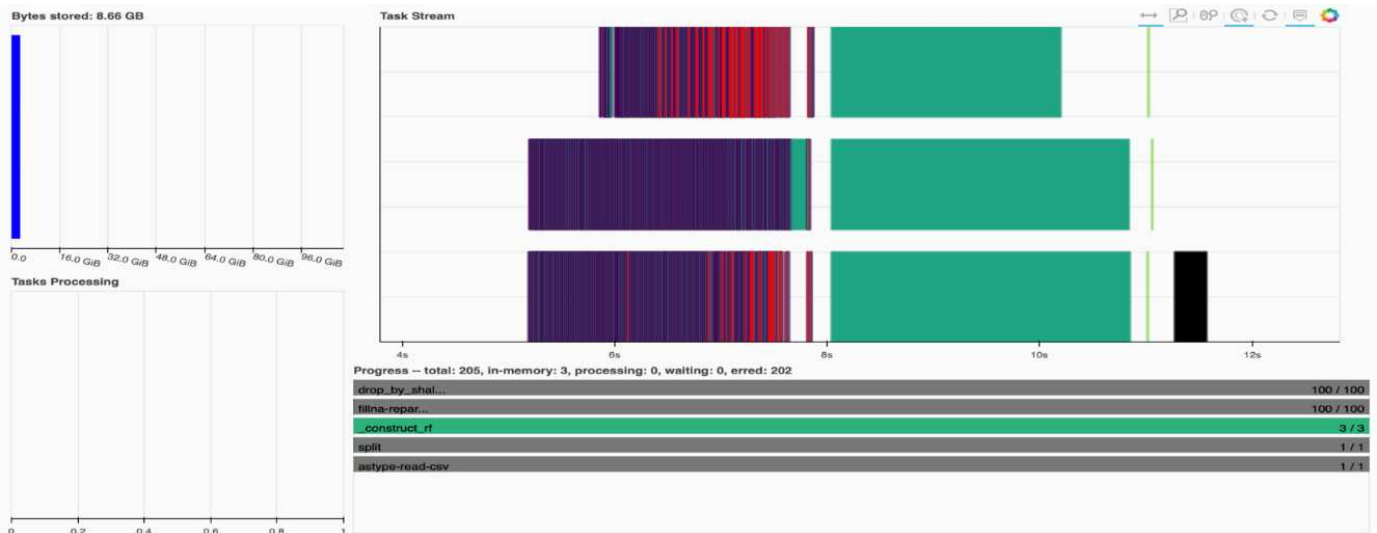
```

使用原生任务流信息板监控 **dask**

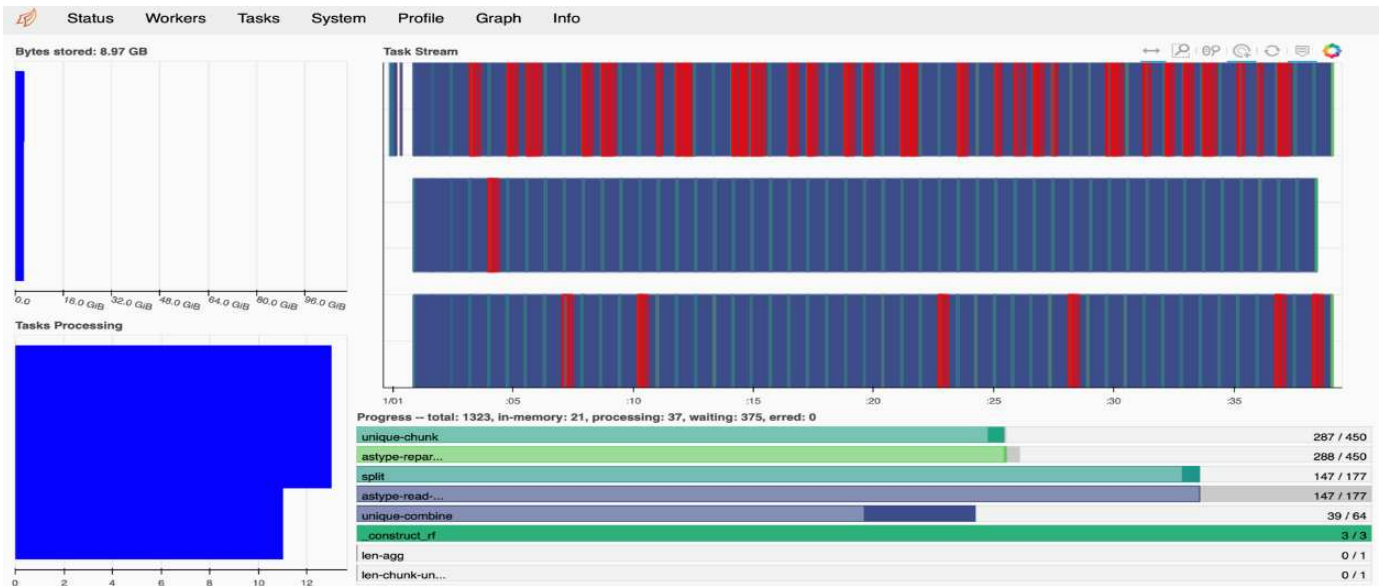
。"dask 分布式计划程序" 以两种形式提供实时反馈：

- 一个交互式信息板，其中包含许多图表和包含实时信息的表
- 一个适合在控制台或笔记本电脑中交互使用的进度条

在我们的案例中，下图显示了如何监控任务进度，包括存储的字节数，详细细分流数量的任务流以及执行关联功能的任务名称的进度。在我们的案例中，由于我们有三个辅助节点，因此流有三个主要区块，而颜色代码表示每个流中的不同任务。



您可以选择分析单个任务并以毫秒为单位检查执行时间，或者确定任何障碍或障碍。例如，下图显示了随机林模型安装阶段的任务流。要执行的功能要多得多，包括用于 DataFrame 处理的唯一区块，用于安装随机林的 `_construct_rf` 等。由于 Criteo Click Logs 中一天的数据非常大（45 GB），因此大部分时间都花在了 DataFrame 操作上。



### 培训时间比较

本节将使用传统熊猫的模型训练时间与 dask 进行比较。对于熊猫，由于处理时间较慢，我们加载的数据较少，以避免内存溢出。因此，我们对结果进行了插值计算，以便进行合理的比较。

下表显示了当熊猫随机林模型（数据集每天 200 亿行中的 5000 万行）所使用的数据明显减少时的原始训练时间比较。15 此示例仅使用所有可用数据的 0.25% 以下。而对于 dask-cuML，我们在所有 200 亿行可用的行上训练了随机林模型。这两种方法的训练时间相当。

方法	培训时间
Scikit 学习：在 15 天仅使用 50 米行作为训练数据	47 分 21 秒
rapids-dask：使用第 15 天的所有 20 B 行作为训练数据	1 小时， 12 分钟和 11 秒

如果我们按线性方式插值训练时间结果，如下表所示，则使用分布式训练和 dask 具有显著优势。传统的熊猫科学学习方法需要 13 天来处理和训练 45 GB 的数据，只需一天的单击日志，而使用快速 dask 方法处理相同数量的数据则要快 262.39 倍。

方法	培训时间
Scikit 学习：使用第 15 天的所有 20 B 行作为训练数据	13 天， 3 小时， 40 分钟和 11 秒
rapids-dask：使用第 15 天的所有 20 B 行作为训练数据	1 小时， 12 分钟和 11 秒

在上表中，您可以看到，通过使用带和 dask 的快速处理功能在多个 GPU 实例之间分布数据处理和模型训练，与使用 scide-Learn 模型训练的传统熊猫 DataFrame 处理相比，运行时间要短得多。此框架支持在云端以及多节点多 GPU 集群内部进行纵向和横向扩展。

### 使用 Prometheus 和 Grafana 监控 dask 和快速流

部署完所有内容后，对新数据运行推断。这些模型可根据浏览活动预测用户是否单击某个



广告。预测结果存储在 dask cuDF 中。您可以使用 Prometheus 监控结果，并在 Grafana 信息板中直观显示结果。

有关详细信息，请参见此 ["RAPID AI 中型 POST"](#)。

使用 **NetApp DataOps** 工具包对数据集和模型进行版本控制

适用于 Kubernetes 的 NetApp DataOps 工具包可将存储资源和 Kubernetes 工作负载抽象到数据科学工作空间级别。这些功能打包在一个简单易用的界面中，专为数据科学家和数据工程师设计。该工具包采用熟悉的 Python 计划形式，可帮助数据科学家和工程师在几秒钟内配置和销毁 JupyterLab 工作空间。这些工作空间可以包含数 TB 甚至数 PB 的存储容量，从而使数据科学家能够将其所有培训数据集直接存储在其项目工作空间中。单独管理工作空间和数据卷的时代已经一去不返。

有关详细信息，请访问工具包 ["GitHub 存储库"](#)。

**Jupyter** 笔记本电脑作为参考

本技术报告涉及两款 Jupyter 笔记本电脑：

- `"* ct-andasrf-colled.ipynb.*"` 此笔记本电脑可从 Criteo TB Click Logs 数据集加载第 15 天的数据，将数据处理并格式化到一个熊猫 DataFrame 中，训练一个 Scikit 学习随机林模型，执行预测并计算准确性。
- `"* Criteo_dask_rf.ipynb.*"` 此笔记本电脑可从 Criteo Terabyte Click Logs 数据集加载第 15 天数据，将数据处理并格式化为 dask cuDF，训练 dask cuML 随机林模型，执行预测并计算准确性。这种分布式数据和模型处理和培训方法通过利用多个具有 GPU 的员工节点来实现高效率。与传统的 ML 方法相比，处理的数据越多，节省的时间就越多。您可以将此笔记本电脑部署在云，内部或混合环境中，其中 Kubernetes 集群包含不同位置的计算和存储，前提是您的网络设置可以自由移动数据和分发型号。

## 结论

Azure NetApp Files，速写和 dask 可通过与 Docker 和 Kubernetes 等业务流程工具集成来加快和简化大规模 ML 处理和培训的部署。通过统一端到端数据管道，此解决方案可降低许多高级计算工作负载固有的延迟和复杂性，从而有效地弥补开发和运营之间的差距。在培训阶段，数据科学家可以对大型数据集运行查询，并与其他用户安全地共享数据和算法模型。

在构建自己的 AI/ML 管道时，配置架构中组件的集成，管理，安全性和可访问性是一项极具挑战性的任务。让开发人员访问和控制其环境也带来了另一组挑战。

通过在云中构建端到端分布式培训模型和数据管道，我们展示了与不利用 GPU 加速数据处理和计算框架的传统开源方法相比，工作流总完成时间有两个显著的提升。

NetApp，Microsoft，开源业务流程框架和 NVIDIA 的结合，将最新技术作为托管服务集于一身，并具有极大的灵活性，可加快技术采用速度，加快新 AI/ML 应用程序的上市速度。这些高级服务在云原生环境中提供，可以轻松地移植到内部部署和混合部署架构中。

## 从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请参见以下资源：

- Azure NetApp Files

- Azure NetApp Files 的解决方案架构页面

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

- 适用于容器的 Trident 持久存储：

- Azure NetApp Files 和 Trident

["https://netapptrident.readthedocs.io/en/stablev20.07/kubernetes/operations/tasks/backends/anf.html"](https://netapptrident.readthedocs.io/en/stablev20.07/kubernetes/operations/tasks/backends/anf.html)

- dask 和 rapids：

- dask

["https://docs.dask.org/en/latest/"](https://docs.dask.org/en/latest/)

- 安装 dask

["https://docs.dask.org/en/latest/install.html"](https://docs.dask.org/en/latest/install.html)

- Dask API

["https://docs.dask.org/en/latest/api.html"](https://docs.dask.org/en/latest/api.html)

- DASK 机器学习

["https://examples.dask.org/machine-learning.html"](https://examples.dask.org/machine-learning.html)

- Dask 分布式诊断

["https://docs.dask.org/en/latest/diagnostics-distributed.html"](https://docs.dask.org/en/latest/diagnostics-distributed.html)

- ML 框架和工具：

- TensorFlow：适用于所有人的开源机器学习框架

["https://www.tensorflow.org/"](https://www.tensorflow.org/)

- Docker

["https://docs.docker.com"](https://docs.docker.com)

- Kubernetes

["https://kubernetes.io/docs/home/"](https://kubernetes.io/docs/home/)

- Kubeflow

["http://www.kubeflow.org/"](http://www.kubeflow.org/)

## 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。运行：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。

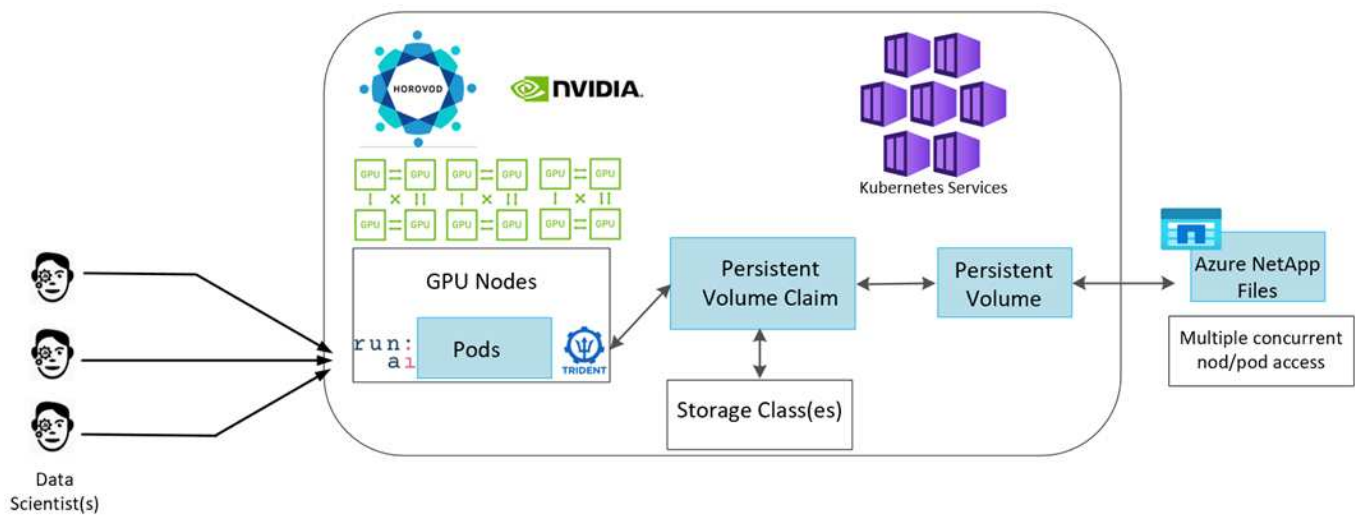
解决方案技术

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 orchestrator** 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。

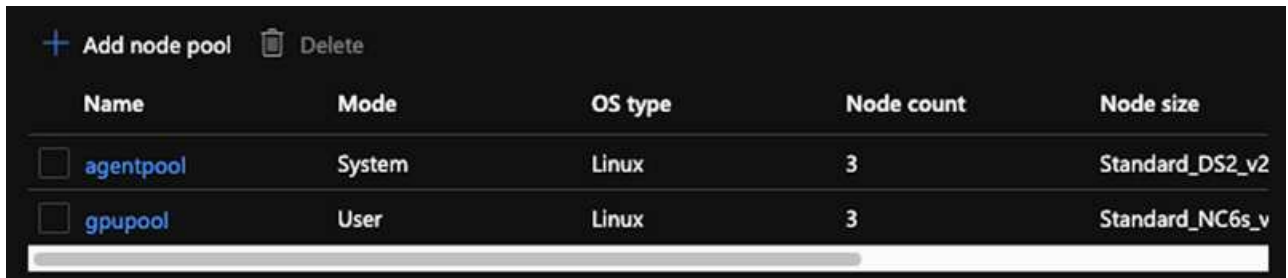
## AK 设置和安装

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：
  - a. 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。



- b. 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。



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：

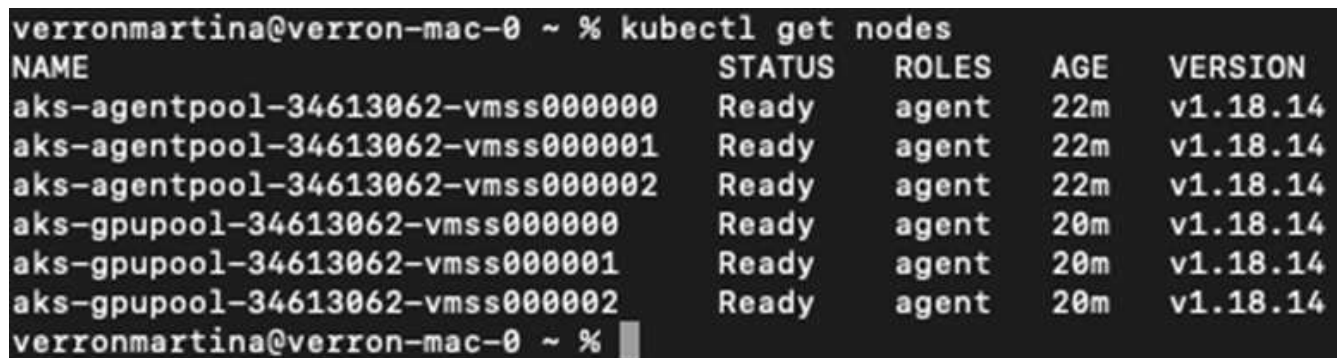
```
az account set --subscription xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx
aks get-credentials --resource-group resourcegroup --name aksclustername
```

- 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。

Microsoft Azure

Search resources, services, and docs (G+/I)

Dashboard > Virtual networks

seanlucelive (Default Directory)

+ Add Manage view Refresh Export to CSV Open query Assign tags Feedback

Filter by name... Subscription == AzureSub01 Resource group == all Location == all Add filter

Showing 1 to 5 of 5 records. No grouping List view

Name ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓
aks-vnet-22885919	MC_sluce_rg_TridentDemo_eastus2	East US 2	AzureSub01

2. 单击子网，然后从顶部工具栏中选择 +Subnet。

Microsoft Azure

Search resources, services, and docs (G+/I)

Dashboard > Virtual networks > aks-vnet-22885919

aks-vnet-22885919 | Subnets

Virtual network

Search (Ctrl+/) + Subnet Gateway subnet Refresh Manage users Delete

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings Address space Connected devices Subnets

Search subnets

Name ↑↓	IPv4 ↑↓	IPv6 (many availab... ↑↓	Delegated to ↑↓	Security group ↑↓
aks-subnet	10.240.0.0/16 (65530 av...	-	-	aks-agentpool-2288591...

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	值或说明#
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 azurenetafiles: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	AGE	NODE	IMAGE	TYPE	PROJECT	USER	GPUs Allocated (Requested)	
Pods Running (Pending)	SERVICE URL(S)								
download-tusimple-data	ContainerCreating	1m	aks-agentpool-34613062-vmss00000a	muneer7589/download-tusimple:1.0	Train	lane-detection	verronmartina	0	0
1	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。

```
kubect1 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										Q		
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-scnn-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
```

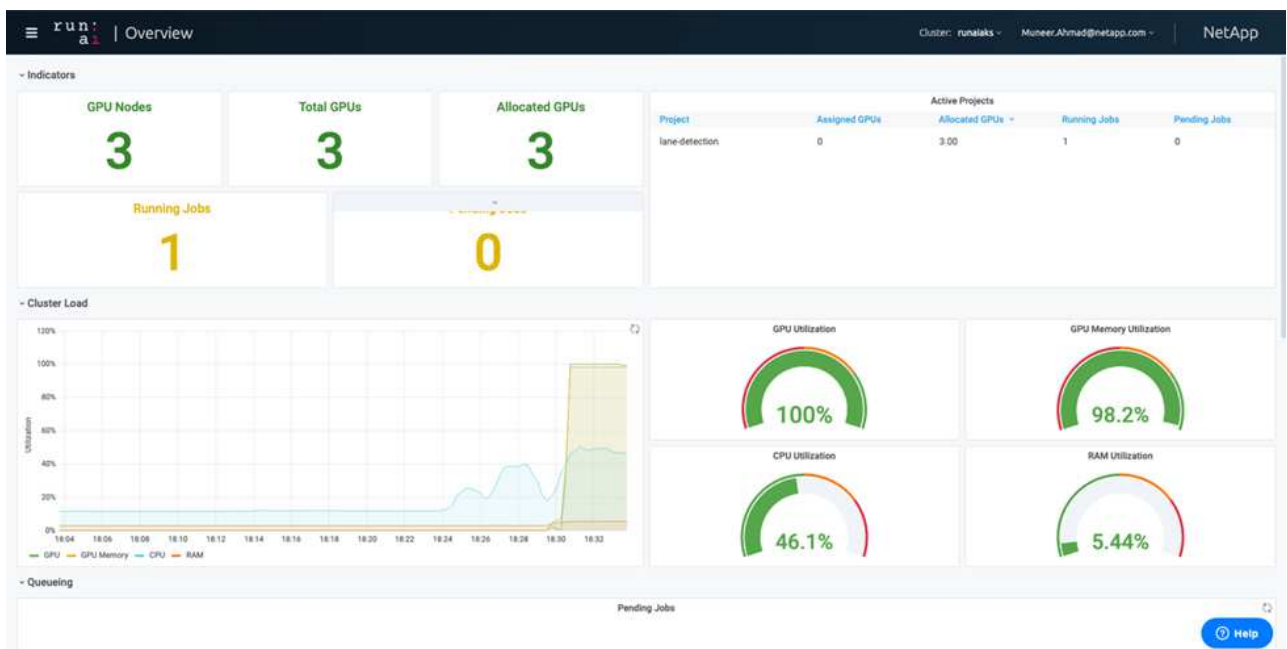
NAME	SERVICE URL(S)	STATUS	AGE	NODE	IMAGE	TYPE	PROJECT	USER	GPUs Allocated (Requested)	PODs
download-tusimple-data		Succeeded	1d		muneer7589/download-tusimple:1.0	Train	lane-detection	verronmartina	0 (0)	0 (0)
dist-lane-detection-training		Init:0/1	2m	<multiple>	muneer7589/dist-lane-detection:3.1	Train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

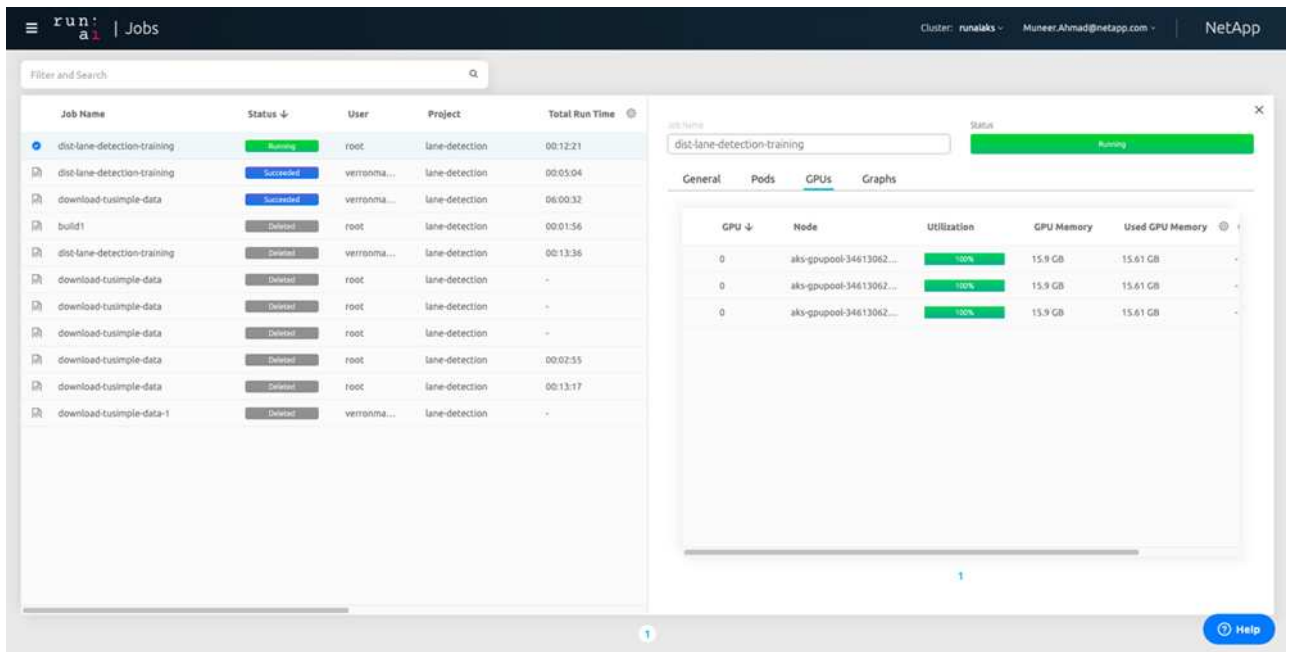
f. 已提交作业日志：

```
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/kubectl 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/kubectl 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 lan-detect-scan-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。

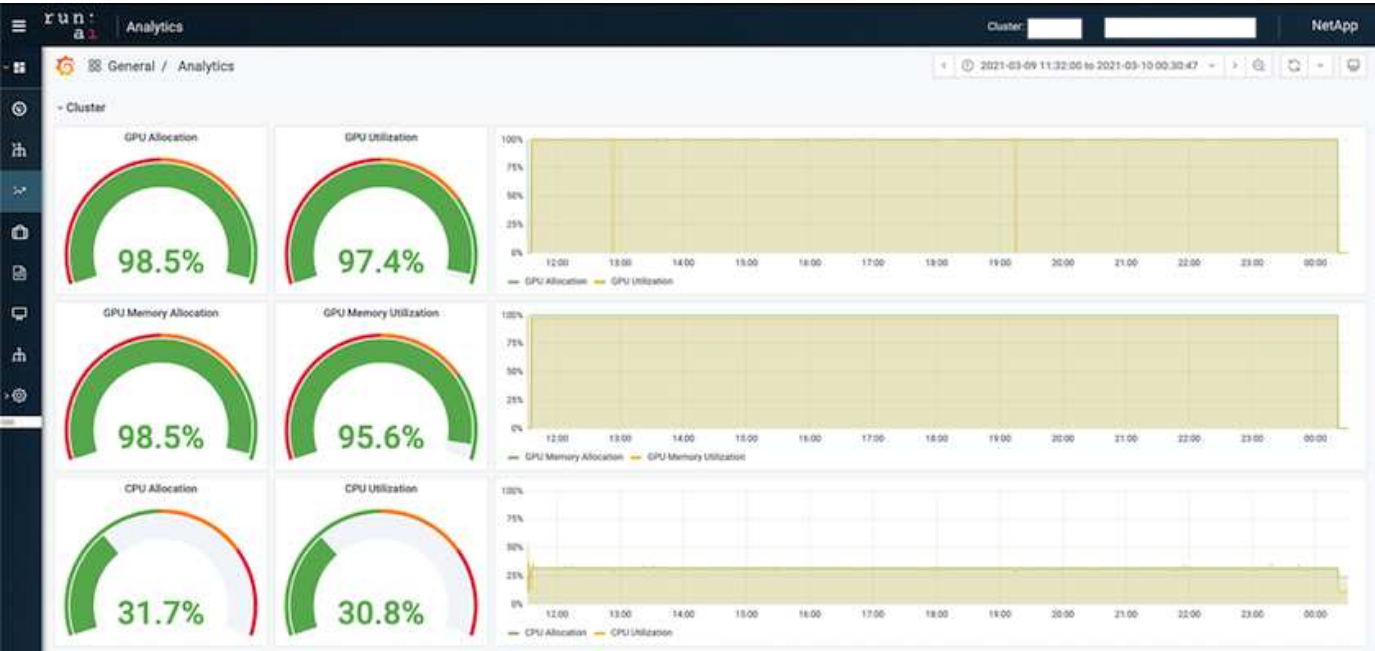
借助解决方案，客户可以从一个小型数据集和几个 GPU 入手。当数据量和 GPU 需求增加时，客户可以动态地横向扩展标准层中的 TB，并快速扩展到高级层，从而在不移动任何数据的情况下获得每 TB 吞吐量的四倍。本

节将进一步介绍此过程。 "Azure NetApp Files 服务级别"。

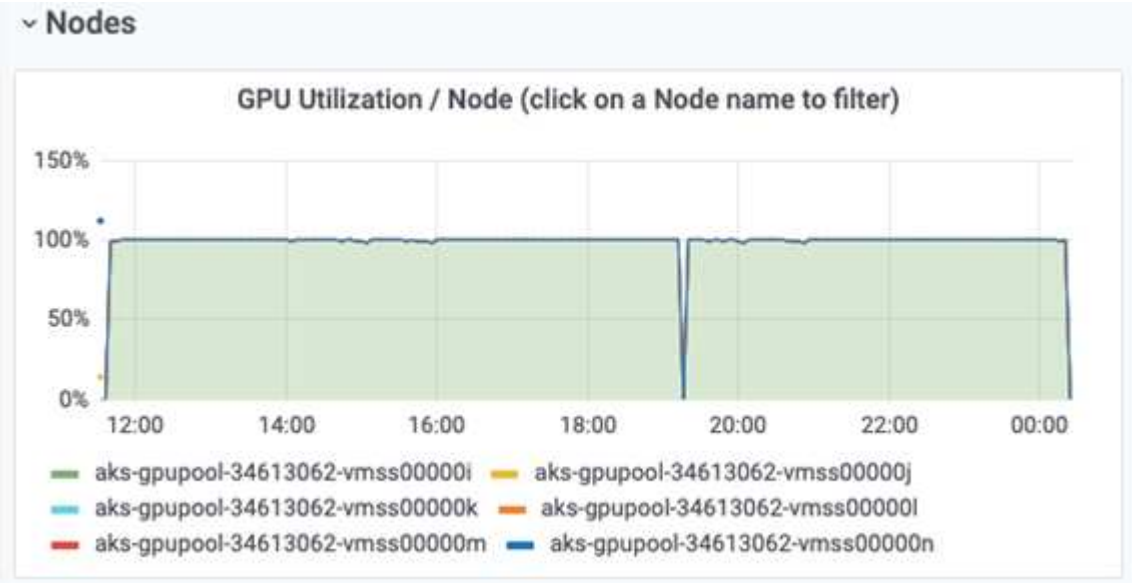
一个 GPU 的处理时间为 12 小时 45 分钟。三个节点上的三个 GPU 的处理时间约为 4 小时 30 分钟。

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。

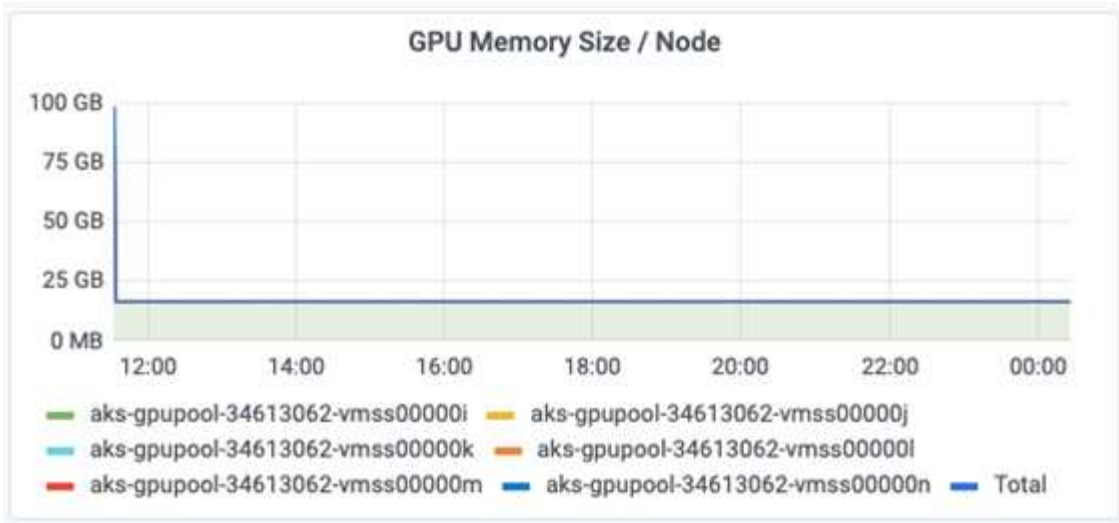


下图显示了单节点 GPU 利用率。

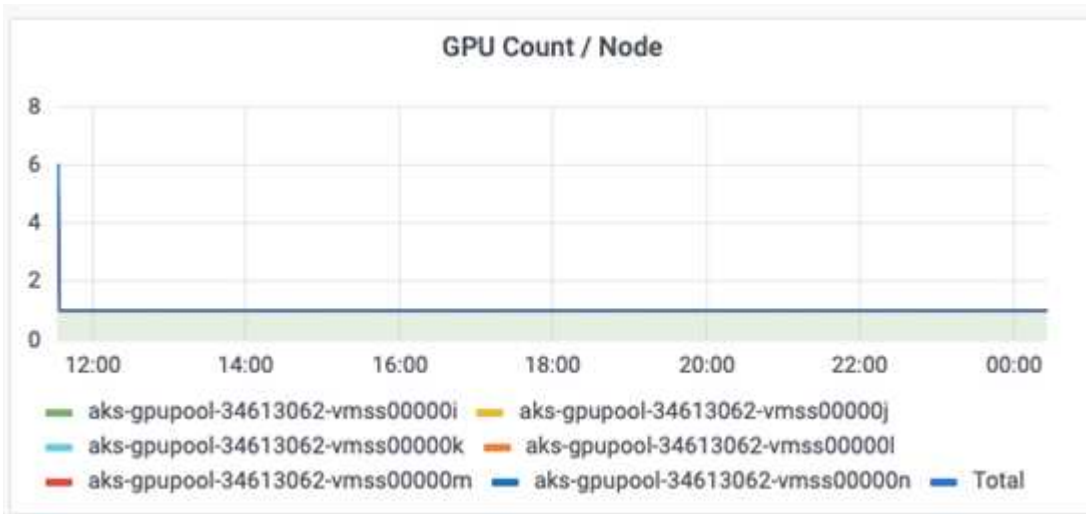


下图显示了单节点内存大小（16 GB）。

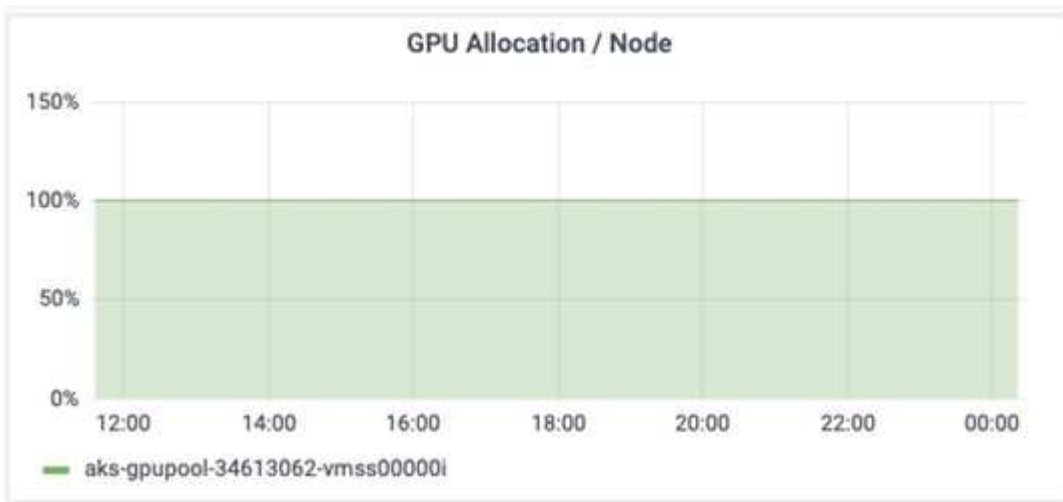




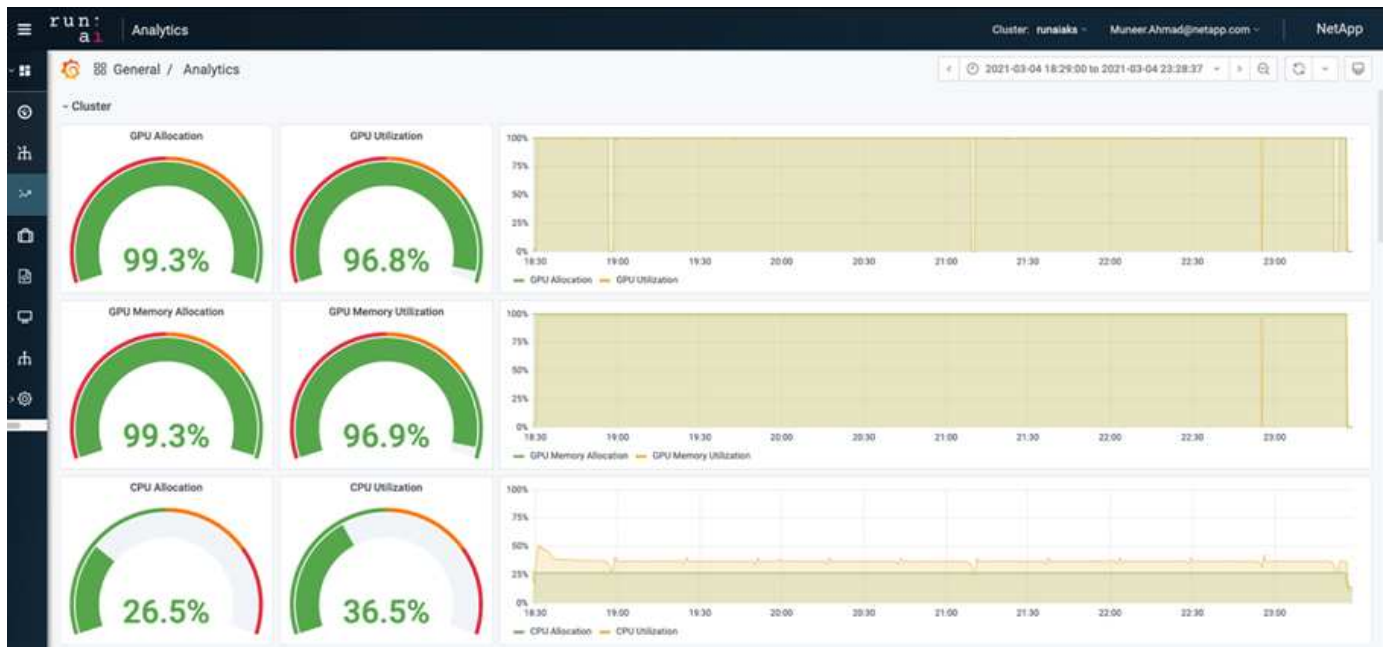
下图显示了单节点 GPU 计数（1）。



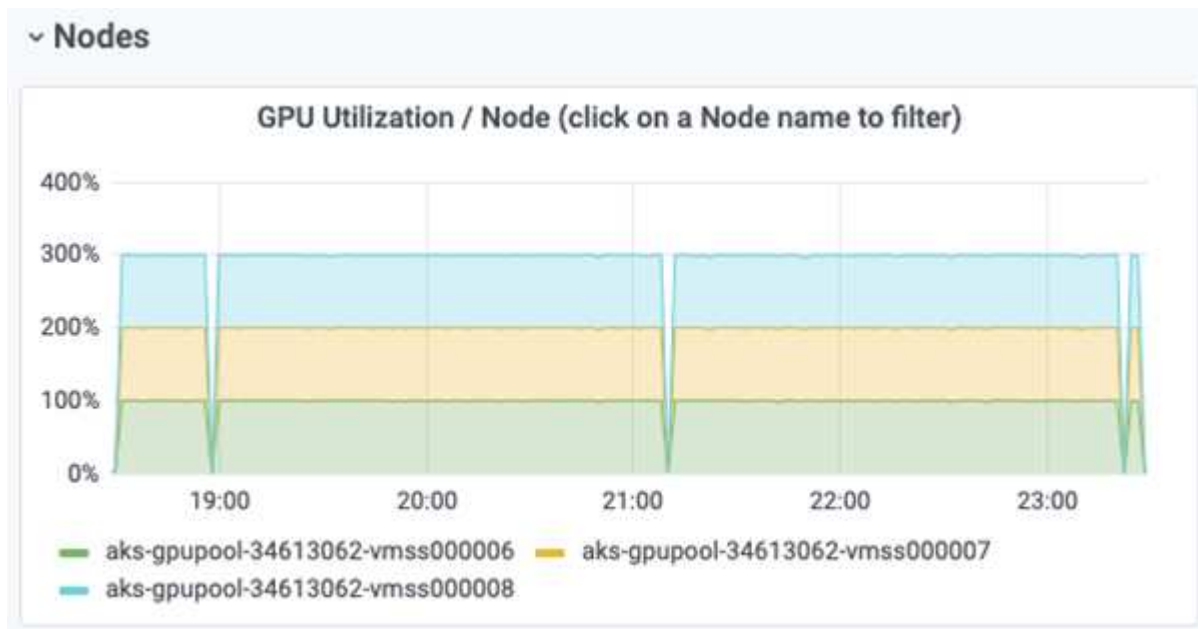
下图显示了单节点 GPU 分配（%）。



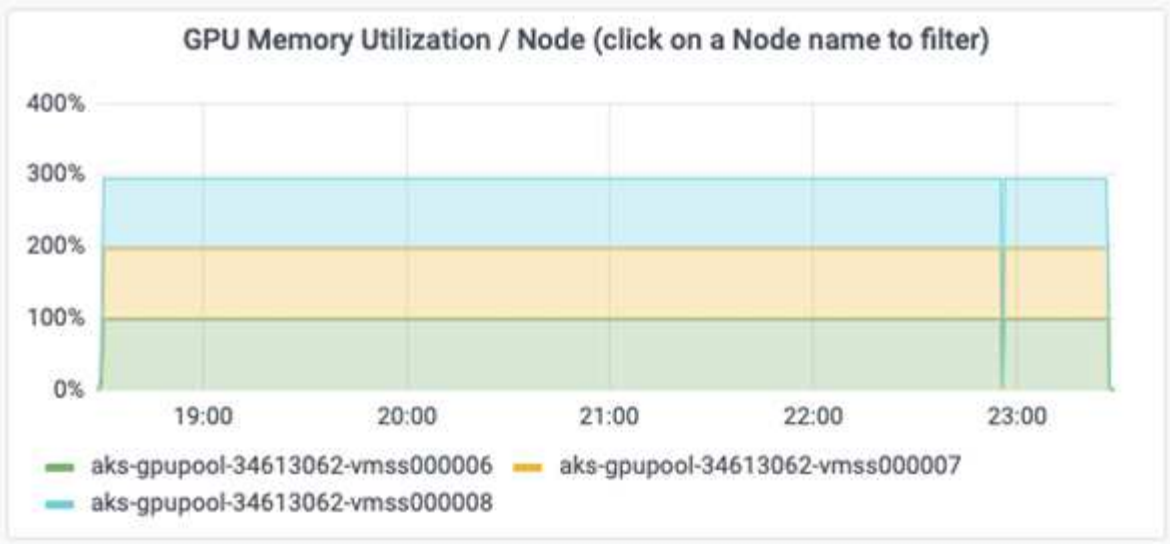
下图显示了三个节点上的三个 GPU — GPU 分配和内存。



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		...

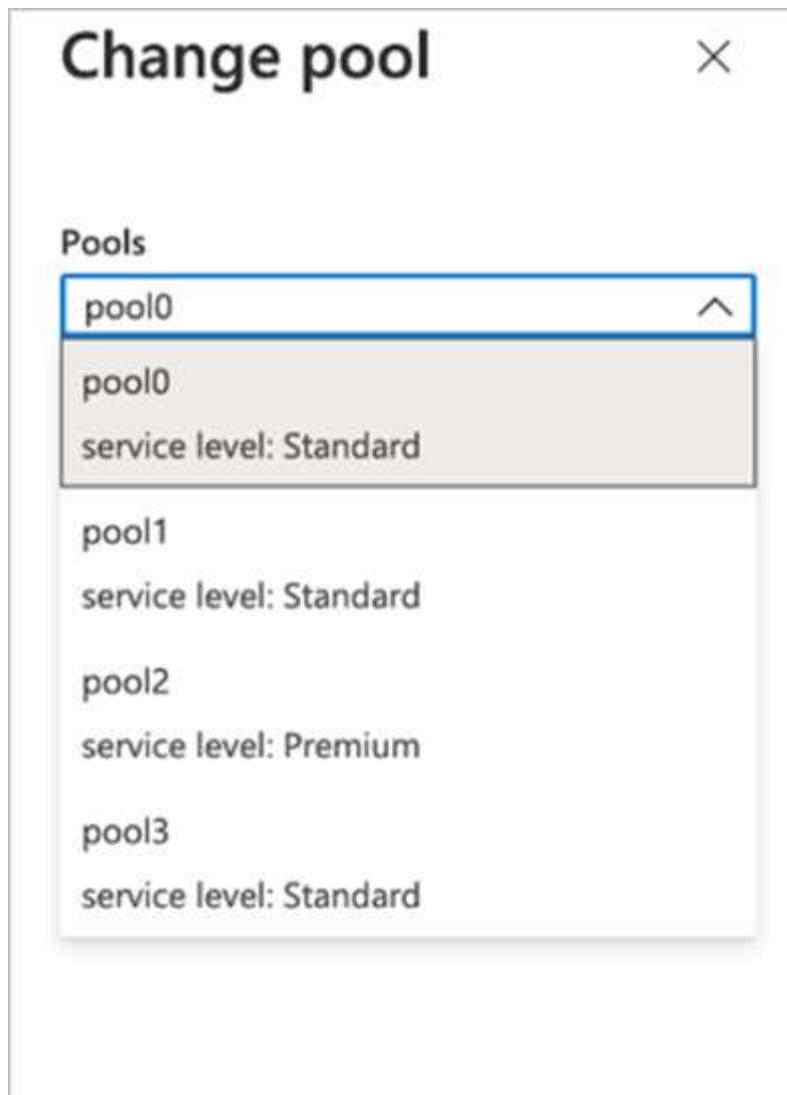
Resize

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 ["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 myacname
-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/)

## TR-4841：采用数据缓存的混合云 AI 操作系统

Rick Huang， David Arnette， NetApp Ychay Ettun， cnvrg.io

数据的爆炸式增长以及 ML 和 AI 的指数级增长已经融合在一起，形成了一个具有独特开发和实施挑战的超字节经济。

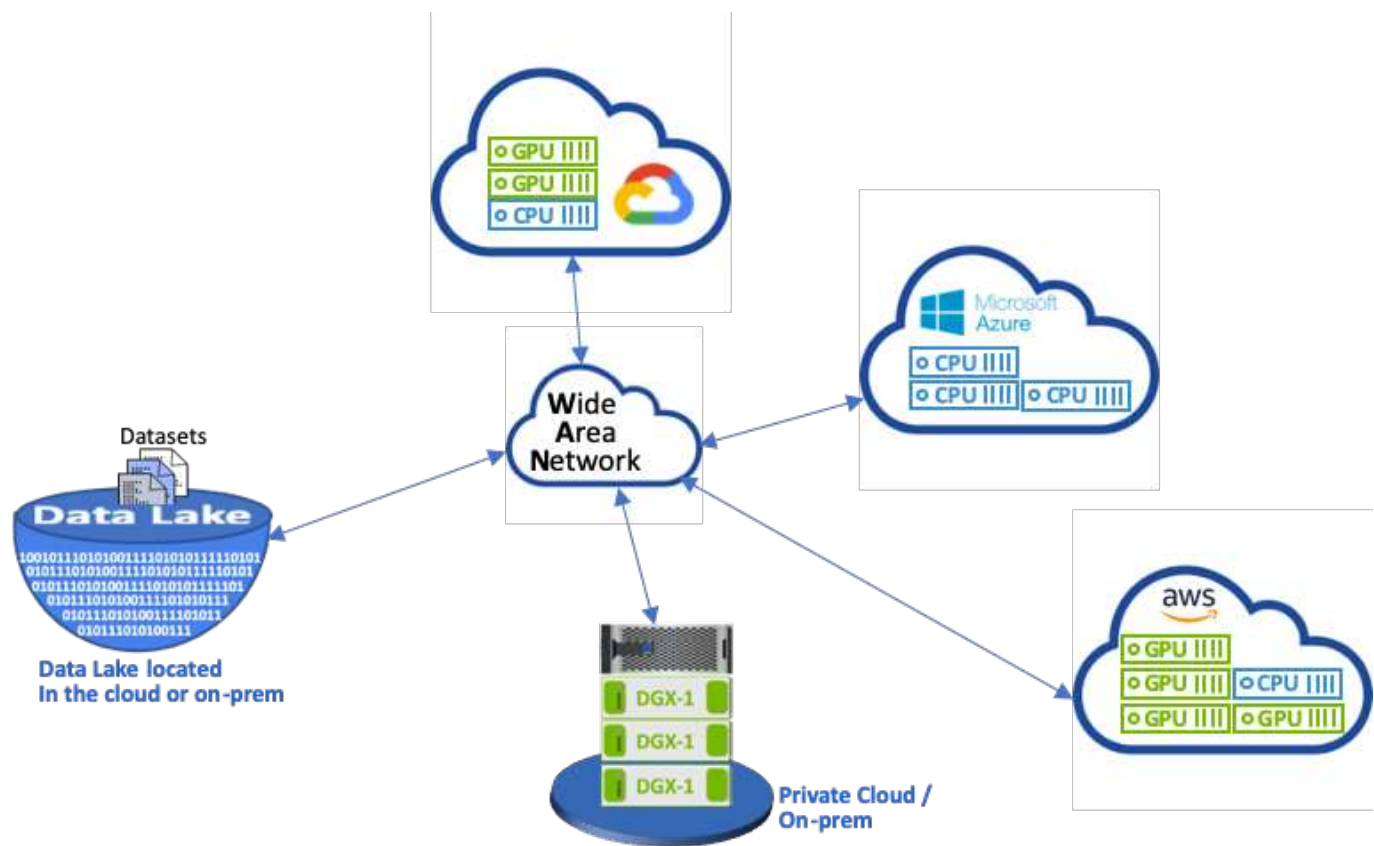
尽管众所周知，ML 模型需要大量数据，并且需要接近计算资源的高性能数据存储，但在实践中，实施这种模型并不是那么直接，尤其是在混合云和弹性计算实例中。大量数据通常存储在低成本数据湖中，GPU 等高性能 AI 计算资源无法高效访问这些数据。在混合云基础架构中，如果某些工作负载在云中运行，而某些工作负载完全位于内部或不同的 HPC 环境中，则此问题会更加严重。

在本文档中，我们介绍了一款全新的解决方案，IT 专业人员 and 数据工程师可以利用可感知拓扑的数据中心创建一个真正的混合云 AI 平台，数据科学家可以利用该平台在计算资源附近即时自动创建数据集缓存。无论它们位于何处。因此，不仅可以完成高性能模型培训，而且还可以带来更多优势，包括多名 AI 实践者的协作，他们可以立即访问数据集版本中心内的数据集缓存，版本和行。

### 用例概述和问题陈述

数据集和数据集版本通常位于数据湖中，例如 NetApp StorageGRID 基于对象的存储，这

样可以降低成本并获得其他运营优势。数据科学家利用这些数据集，通过多个步骤对其进行设计，使其为使用特定模型进行培训做好准备，通常会在整个过程中创建多个版本。下一步，数据科学家必须选择经过优化的计算资源（ GPU ，高端 CPU 实例，内部集群等）来运行此模型。下图显示了 ML 计算环境中数据集不接近的情况。



但是，多个训练实验必须在不同的计算环境中并行运行，每个实验都需要从数据湖中下载数据集，这是一个昂贵且耗时的过程。无法保证数据集与计算环境（尤其是混合云）的距离。此外，使用同一数据集运行自己实验的其他团队成员也必须经历同样艰巨的过程。除了明显缓慢的数据访问之外，还存在一些挑战，包括跟踪数据集版本，数据集共享，协作和可重现性方面的困难。

客户要求

为了在高效利用资源的同时实现高性能 ML 运行，客户要求可能有所不同；例如，客户可能需要满足以下要求：

- 从执行训练模型的每个计算实例快速访问数据集，而不会导致昂贵的下载和复杂的数据访问
- 在云或内部环境中使用任何计算实例（ GPU 或 CPU ），而无需考虑数据集的位置
- 通过在同一数据集中与不同计算资源并行运行多个训练实验，而不会出现不必要的延迟和数据延迟，提高了效率和工作效率
- 最大限度地降低计算实例成本
- 利用工具来保留数据集，其沿革，版本和其他元数据详细信息的记录，从而提高了可重现性
- 增强了共享和协作功能，使团队中的任何授权成员都可以访问数据集并运行实验

要使用 NetApp ONTAP 数据管理软件实施数据集缓存，客户必须执行以下任务：

- 配置和设置最接近计算资源的 NFS 存储。



- 确定要缓存的数据集和版本。
- 监控提交到缓存数据集的总内存以及可用于其他缓存提交的 NFS 存储容量（例如缓存管理）。
- 如果数据集在特定时间未使用，则会使其过期。默认值为一天；其他配置选项可用。

## 解决方案概述

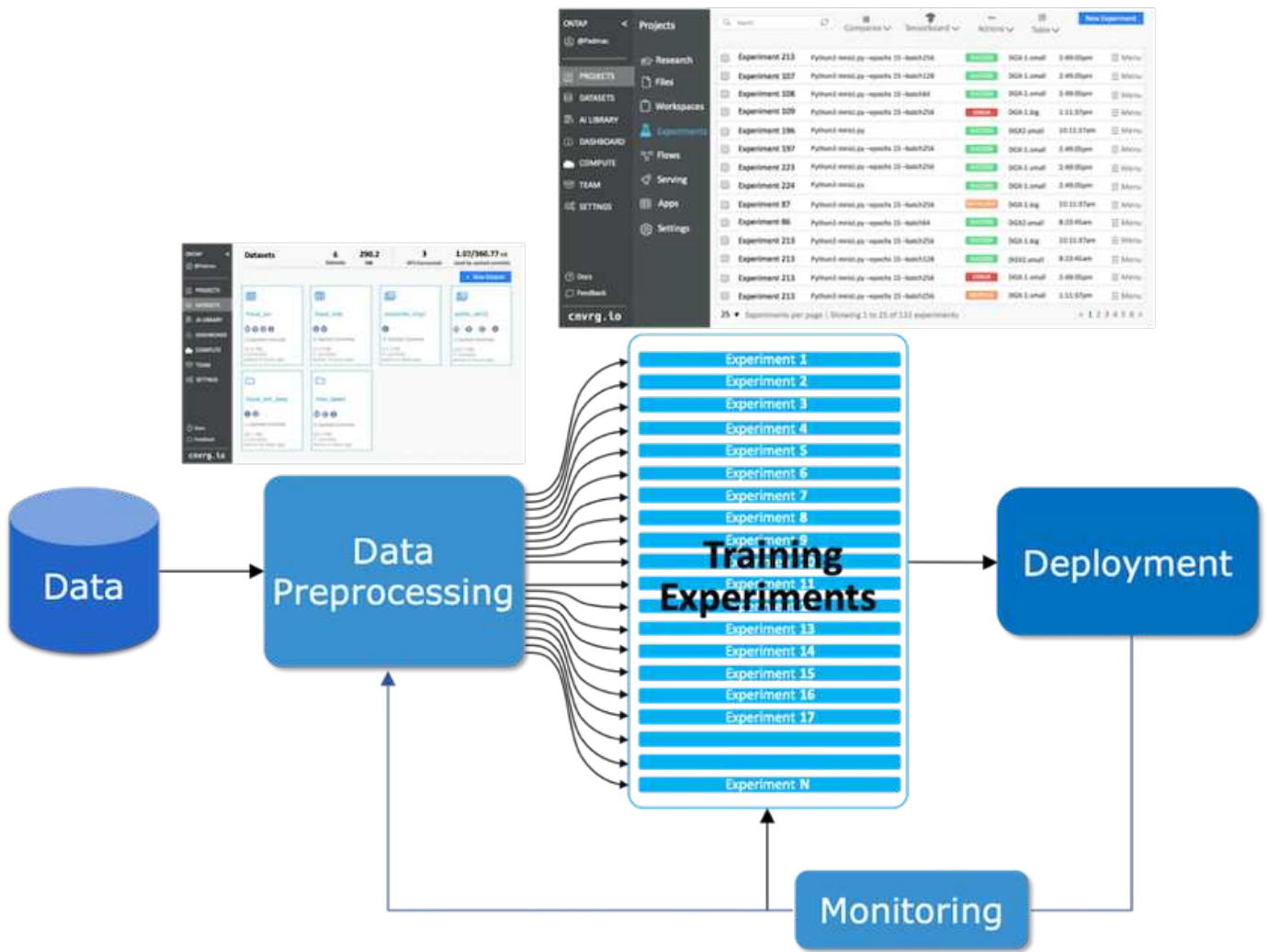
本节将介绍传统数据科学管道及其缺点。此外，还介绍了建议的数据集缓存解决方案的架构。

### 传统数据科学管道和缺点

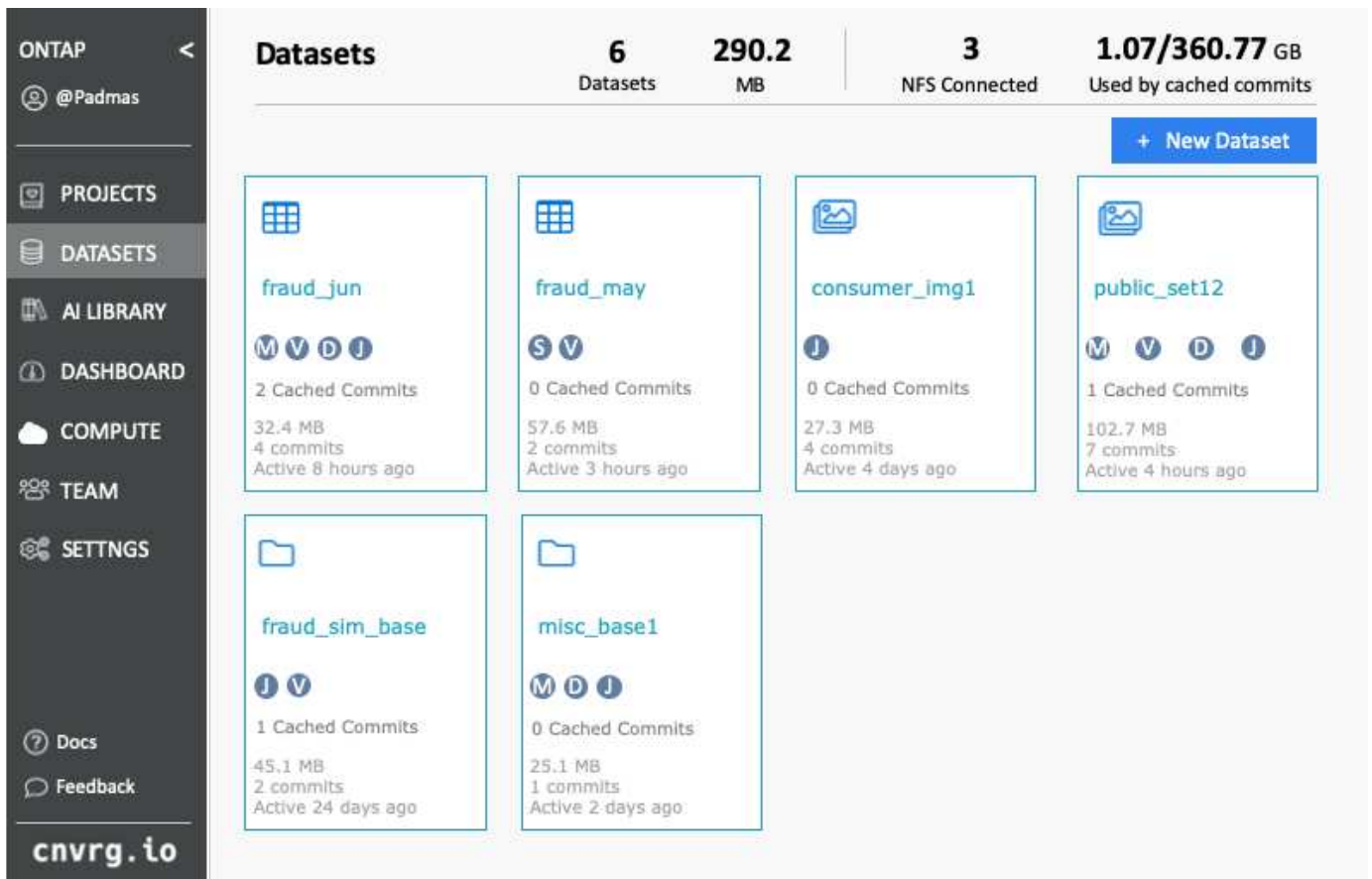
ML 模型开发和部署的典型顺序涉及以下迭代步骤：

- 正在载入数据
- 数据预处理（创建多个版本的数据集）
- 运行多个涉及超参数优化，不同型号等的实验
- 部署
- Monitoringcnvrg.io 开发了一个全面的平台，可以自动执行从研究到部署的所有任务。下图显示了与管道相关的一小部分信息板屏幕截图。





从公有存储库和私有数据中使用多个数据集非常常见。此外，每个数据集可能具有多个版本，这些版本是由数据集清理或功能工程产生的。需要一个信息板来提供数据集中心和版本中心，以确保团队可以使用协作和一致性工具，如下图所示。



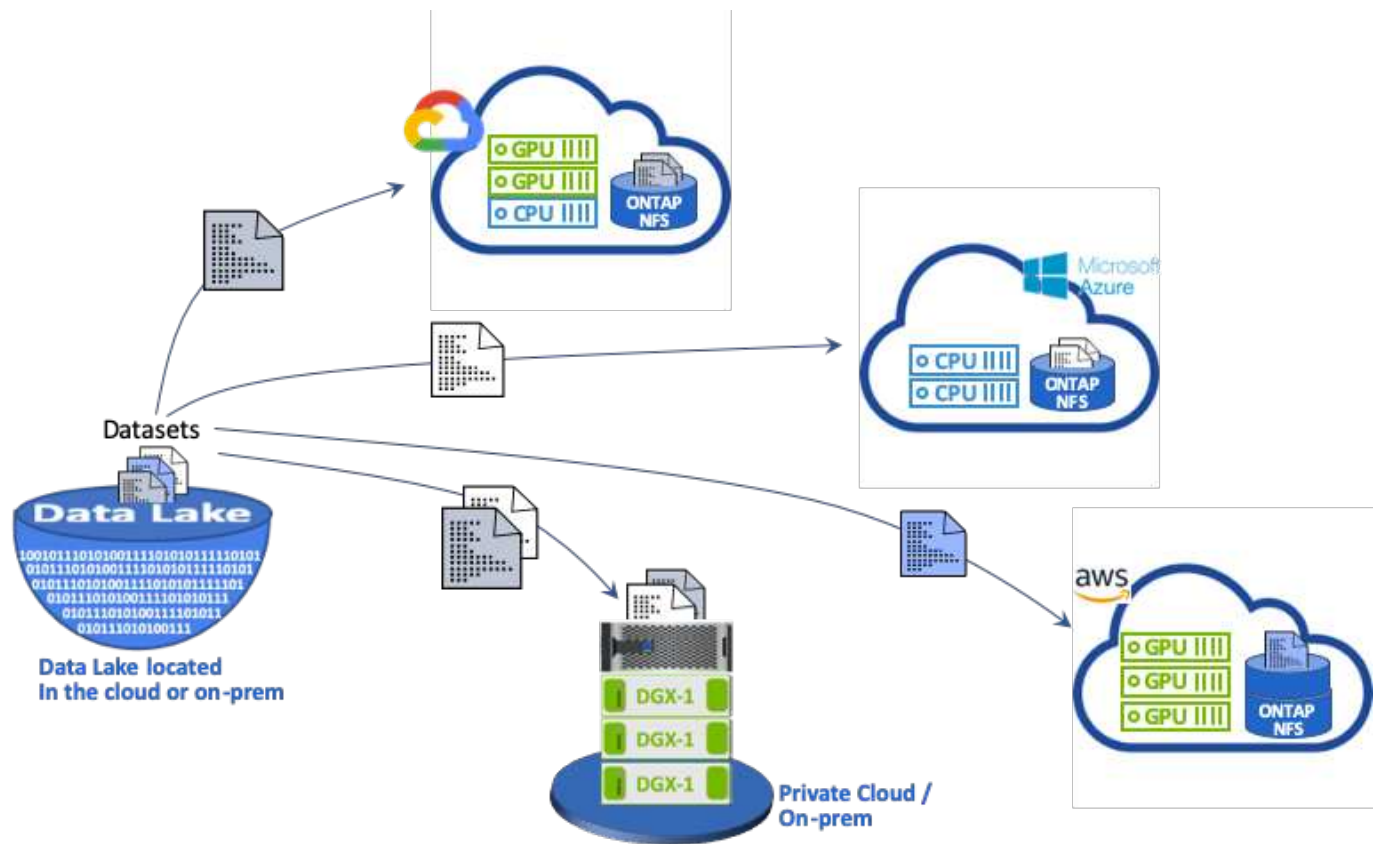
管道的下一步是培训，这需要多个并行的培训模型实例，每个实例都与一个数据集和一个特定计算实例相关联。将数据集绑定到使用特定计算实例的特定实验是一项挑战，因为某些实验可能由 Amazon Web Services（AWS）中的 GPU 实例执行，而其他实验则由内部 DGX-1 或 DGX-2 实例执行。可能会在 GCP 的 CPU 服务器中执行其他实验，但数据集位置与执行培训的计算资源不是很近。如果距离合理，则从数据集存储到计算实例的连接将达到全 10GbE 或更高的低延迟。

数据科学家通常会将数据集下载到执行培训和实验的计算实例中。但是，此方法可能会出现以下几个问题：

- 当数据科学家将数据集下载到计算实例时，无法保证集成计算存储具有高性能（高性能系统的一个示例是 ONTAP AFF A800 NVMe 解决方案）。
- 如果下载的数据集驻留在一个计算节点中，则在多个节点上执行分布式模型时，存储可能会成为瓶颈（与 NetApp ONTAP 高性能分布式存储不同）。
- 由于队列冲突或优先级问题，下次迭代训练实验可能会在不同的计算实例中执行，这再次导致从数据集到计算位置的网络距离过长。
- 在同一计算集群上执行训练实验的其他团队成员不能共享此数据集；每个团队成员都从任意位置执行数据集（昂贵的）下载。
- 如果后续培训作业需要使用同一数据集的其他数据集或版本，则数据科学家必须再次将数据集（昂贵）下载到执行 training.NetApp 的计算实例中，而 cnvrg.io 已创建一个新的数据集缓存解决方案来消除这些障碍。解决方案通过在 ONTAP 高性能存储系统上缓存热数据集，加快了 ML 管道的执行速度。使用 ONTAP NFS 时，数据集会在由 NetApp 提供支持的数据网络结构（例如 AFF A800）中缓存一次（并且只缓存一次），该数据网络结构与计算搭配使用。由于 NetApp ONTAP NFS 高速存储可以为多个 ML 计算节点提供服务，因此培训模型的性能得到了优化，从而为企业节省了成本，提高了工作效率并提高了运营效率。

NetApp 和 cnvrg.io 提供的此解决方案可提供数据集缓存，如下图所示。通过数据集缓存，数据科学家可以选择所需的数据集或数据集版本，并将其移动到靠近 ML 计算集群的 ONTAP NFS 缓存中。现在，数据科学家可以运行多个实验，而不会造成延迟或下载。此外，所有协作工程师都可以将同一数据集与连接的计算集群结合使用（并可自由选择任何节点），而无需从数据湖中进行额外下载。数据科学家可以获得一个信息板，用于跟踪和监控所有数据集和版本，并查看缓存的数据集。

cnvrg.io 平台会自动检测某个时间内未使用的过期数据集，并从缓存中将其转出，从而为更常用的数据集保留可用的 NFS 缓存空间。需要注意的是，使用 ONTAP 的数据集缓存可在云端和内部环境中运行，从而提供最大的灵活性。



## 概念和组件

本节介绍与 ML 工作流中的数据缓存相关的概念和组件。

### 机器学习

对于全球许多企业和组织来说，ML 正迅速变得至关重要。因此，IT 和 DevOps 团队现在面临着标准化 ML 工作负载以及配置云，内部和混合计算资源的挑战，这些资源支持 ML 作业和管道所需的动态密集型工作流。

### 基于容器的机器学习和 Kubernetes

容器是在共享主机操作系统内核上运行的隔离用户空间实例。容器的采用率正在快速增长。容器可提供许多与虚拟机（VM）相同的应用程序沙盒优势。但是，由于虚拟机所依赖的虚拟机管理程序和子操作系统层已被消除，因此容器的重量要轻得多。

此外，还可以通过容器直接将应用程序依赖关系，运行时间等内容高效地打包到应用程序中。最常用的容器打包

格式是 Docker 容器。已采用 Docker 容器格式进行容器化的应用程序可以在可以运行 Docker 容器的任何计算机上执行。即使计算机上不存在应用程序的依赖关系，也是如此，因为所有依赖关系都打包在容器中。有关详细信息，请访问 "[Docker 网站](#)"。

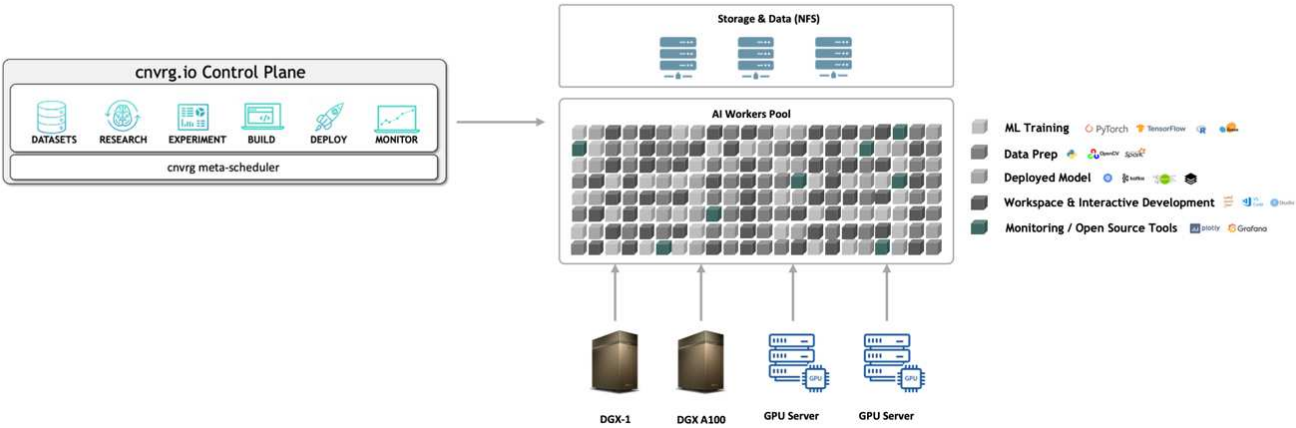
Kubernetes 是一款广受欢迎的容器编排程序，可帮助数据科学家启动基于容器的灵活作业和管道。它还支持基础架构团队在一个受管云原生环境中管理和监控 ML 工作负载。有关详细信息，请访问 "[Kubernetes 网站](#)"。

**cnvrg.io**

cnvrg.io 是一款 AI 操作系统，可将企业管理，扩展和加速 AI 和数据科学开发的方式从研究转变为生产。代码优先平台由数据科学家为数据科学家构建，可灵活地在内部或云中运行。借助模型管理，MLOps 和持续的 ML 解决方案，cnvrg.io 为数据科学团队带来了一流的技术，因此他们可以减少在开发运营上花费的时间，专注于真正的魔力—算法。自使用 cnvrg.io 以来，各个行业的团队已获得更多生产模式，从而增加了业务价值。

**cnvrg.io 元数据计划程序**

cnvrg.io 具有一个独特的架构，允许 IT 和工程师将不同的计算资源连接到同一控制平面，并使 cnvrg.io 管理所有资源中的 ML 作业。这意味着它可以连接多个内部 Kubernetes 集群，VM 服务器和云帐户，并在所有资源上运行 ML 工作负载，如下图所示。



**cnvrg.io 数据缓存**

借助 cnvrg.io，数据科学家可以利用其数据缓存技术定义热数据集和冷数据集版本。默认情况下，数据集存储在集中式对象存储数据库中。然后，数据科学家可以在选定计算资源上缓存特定数据版本，以节省下载时间，从而提高 ML 开发和工作效率。已缓存且在几天内未使用的数据集将自动从选定 NFS 中清除。只需单击一下鼠标即可执行缓存和清除操作；无需进行编码，IT 或 DevOps 工作。

**cnvrg.io 流和 ML 管道**

cnvrg.io 流是一种用于构建生产 ML 管道的工具。流中的每个组件都是一个脚本 / 代码，在选定计算上运行，并具有一个基本 Docker 映像。这种设计使数据科学家和工程师能够构建一个可同时在内部和云中运行的管道。cnvrg.io 可确保数据，参数和项目在不同组件之间移动。此外，还会对每个流进行监控和跟踪，以实现 100% 可重现的数据科学。

**cnvrg.io 核心**

cnvrg.io 核心是数据科学社区的一个免费平台，可帮助数据科学家更加专注于数据科学，而不是专注于开发运营。核心灵活的基础架构使数据科学家能够控制使用任何语言，AI 框架或计算环境，无论是内部环境还是云环境，以便他们能够做到最擅长的事情，构建算法。在任何 Kubernetes 集群上，只需一个命令即可轻松安装



cnvrg-io 核心。

## NetApp ONTAP AI

ONTAP AI 是一款适用于 ML 和深度学习（DL）工作负载的数据中心参考架构，它使用 NetApp AFF 存储系统和采用 Tesla V100 GPU 的 NVIDIA DGX 系统。ONTAP AI 基于基于 100 Gb 以太网的行业标准 NFS 文件协议，可为客户提供高性能 ML/DL 基础架构，该基础架构使用标准数据中心技术来降低实施和管理开销。通过使用标准化网络和协议，ONTAP AI 可以集成到混合云环境中，同时保持操作的一致性和精简性。作为一款经过预先验证的基础架构解决方案，ONTAP AI 可减少部署时间和风险，并显著降低管理开销，从而使客户能够更快地实现价值。

## NVIDIA DeepOps

DeepOps 是 NVIDIA 的一个开源项目，通过使用 Ansible，可以根据最佳实践自动部署 GPU 服务器集群。DeepOps 采用模块化设计，可用于执行各种部署任务。在本文档及其所述的验证练习中，DeepOps 用于部署一个由 GPU 服务器辅助节点组成的 Kubernetes 集群。有关详细信息，请访问 ["DeepOps 网站"](#)。

## NetApp Trident

Trident 是一款由 NetApp 开发和维护的开源存储编排程序，可大大简化 Kubernetes 工作负载的永久性存储的创建，管理和使用。Trident 本身是 Kubernetes 的本机应用程序，它直接在 Kubernetes 集群中运行。借助 Trident，Kubernetes 用户（开发人员，数据科学家，Kubernetes 管理员等）可以采用他们已熟悉的标准 Kubernetes 格式创建，管理永久性存储卷并与其交互。同时，他们还可以利用 NetApp 的高级数据管理功能以及由 NetApp 技术支持的数据网络结构。Trident 可将持久存储的复杂性抽象化，并使其易于使用。有关详细信息，请访问 ["Trident 网站"](#)。

## NetApp StorageGRID

NetApp StorageGRID 是一款软件定义的对象存储平台，旨在通过提供简单的类似于云的存储来满足这些需求，用户可以使用 S3 协议访问这些存储。StorageGRID 是一种横向扩展系统，旨在支持互联网连接站点之间的多个节点，而不管距离如何。借助 StorageGRID 的智能策略引擎，用户可以选择跨站点的纠删编码对象，以便在远程站点之间实现地理故障恢复能力或对象复制，从而最大程度地减少 WAN 访问延迟。StorageGRID 在此解决方案中提供了一个出色的私有云主对象存储数据湖。

## NetApp Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP 数据管理软件可以灵活地为用户数据提供控制，保护和效率，同时还可以灵活地使用 AWS，Google 云平台 and Microsoft Azure 等公有云提供商。Cloud Volumes ONTAP 是一款基于 NetApp ONTAP 存储软件构建的云原生数据管理软件，可为用户提供出色的通用存储平台来满足其云数据需求。在云端和内部部署中使用相同的存储软件，可以为用户提供 Data Fabric 的价值，而无需培训 IT 员工掌握全新的数据管理方法。

对于对混合云部署模式感兴趣的客户，Cloud Volumes ONTAP 可以在大多数公有云中提供相同的功能和同类领先的性能，以便在任何环境中提供一致，无缝的用户体验。

## 硬件和软件要求

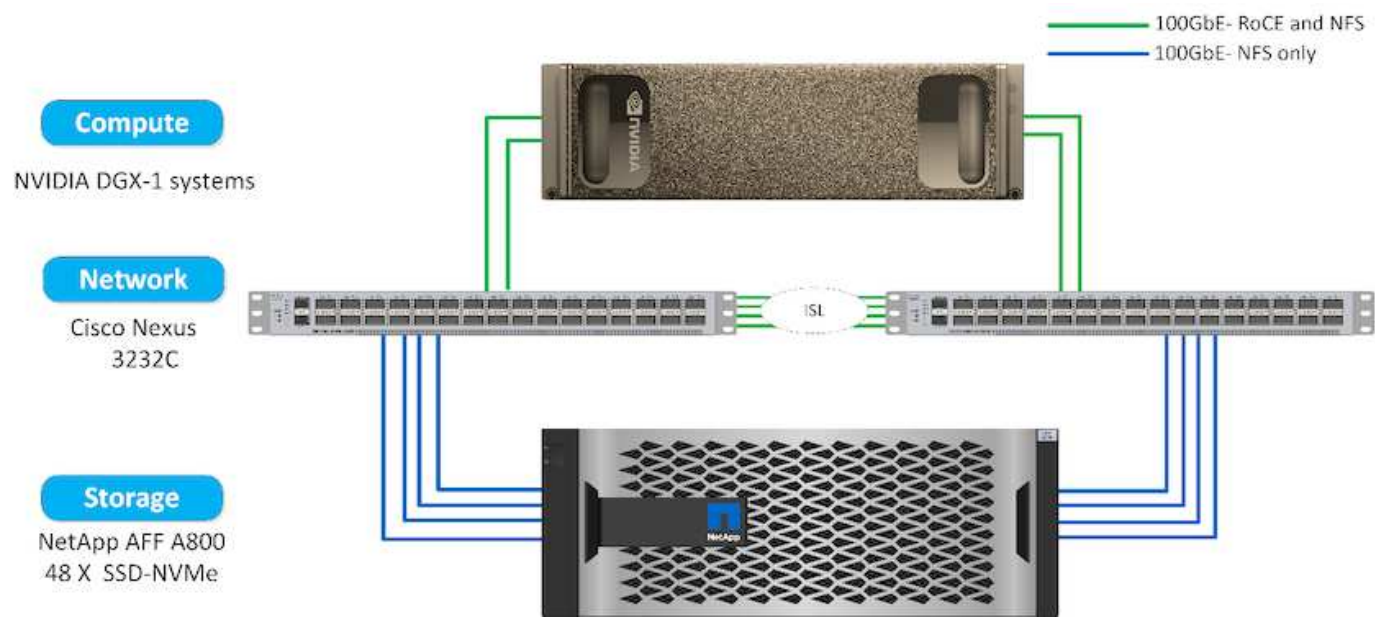
本节介绍 ONTAP AI 解决方案的技术要求。

### 硬件要求

虽然硬件要求取决于特定的客户工作负载，但 ONTAP AI 可以在任何规模部署，用于数据工程，模型培训和生产推理，从单个 GPU 到机架级配置，用于大规模 ML/DL 操作。有关 ONTAP AI 的详细信息，请参见 ["ONTAP](#)

AI 网站"。

此解决方案已通过使用 DGX-1 系统进行计算，使用 NetApp AFF A800 存储系统和使用 Cisco Nexus 3232C 进行网络连接的验证。在此验证中使用的 AFF A800 可支持多达 10 个 DGX-1 系统，用于大多数 ML/DL 工作负载。下图显示了此验证中用于模型培训的 ONTAP AI 拓扑。



要将此解决方案扩展到公有云，可以将 Cloud Volumes ONTAP 与云 GPU 计算资源一起部署，并集成到混合云数据网络结构中，使客户能够使用适合任何给定工作负载的任何资源。

软件要求

下表显示了此解决方案验证中使用的特定软件版本。

组件	version
Ubuntu	18.04.4 LTS
NVIDIA DGX 操作系统	4.4.0
NVIDIA DeepOps	20.02.1
Kubernetes	1.15
掌舵	3.1.0
cnvrg.io	3.0.0
NetApp ONTAP	9.6P4

在此解决方案验证中，Kubernetes 会在 DGX-1 系统上部署为单节点集群。对于大规模部署，应部署独立的 Kubernetes 主节点，以提供高可用性的管理服务，并为 ML 和 DL 工作负载预留有价值的 DGX 资源。

解决方案部署和验证详细信息

以下各节将讨论解决方案部署和验证的详细信息。

## ONTAP AI 部署

部署 ONTAP AI 需要安装和配置网络，计算和存储硬件。本文档不会介绍有关部署 ONTAP AI 基础架构的具体说明。有关详细的部署信息，请参见 ["NVA-1121-Deploy：由 NVIDIA 提供支持的 NetApp ONTAP AI"](#)。

在此解决方案验证中，创建了一个卷并将其挂载到 DGX-1 系统。然后，该挂载点会挂载到容器中，以使数据可供训练访问。对于大规模部署，NetApp Trident 可自动创建和挂载卷，以消除管理开销并支持最终用户管理资源。

## Kubernetes 部署

要使用 NVIDIA DeepOps 部署和配置 Kubernetes 集群，请从部署跳转主机执行以下任务：

1. 按照上的说明下载 NVIDIA DeepOps ["Getting Started 页面"](#) 在 NVIDIA DeepOps GitHub 站点上。
2. 按照上的说明在集群中部署 Kubernetes ["《 Kubernetes 部署指南》"](#) 在 NVIDIA DeepOps GitHub 站点上。



要使 DeepOps Kubernetes 部署正常工作，所有 Kubernetes 主节点和工作节点上必须存在相同的用户。

如果部署失败，请在 `deepops/config/group_vars/K8s-cluster.yml` 中将 `kubectll_localhost` 的值更改为 `false`，然后重复步骤 2。Copy kubectll binary to Ansible host 任务仅在 `kubectll_localhost` 值为 `true` 时执行，它依赖于 Fetch Ansible 模块，该模块存在已知的内存使用问题。这些内存使用问题有时可能会使任务发生原因失败。如果任务因内存问题描述而失败，则部署操作的其余部分将无法成功完成。

如果在将 `kubectll_localhost` 的值更改为 `false` 后部署成功完成，则必须手动将 `kubectll` 二进制文件从 Kubernetes 主节点复制到部署跳转主机。您可以通过直接在特定主节点上运行 `which kubectll` 命令来查找 `kubectll` 二进制文件在该节点上的位置。

## cnvrg.io 部署

本节提供了使用 Helm 图表部署 cnvrg 核心的详细信息。

### 使用 Helm 部署 cnvrg 核心

使用任何集群，内部，Minikube 或任何云集群（如 AKS，EKS 和 GKE），Helm 是快速部署 cnvrg 的最简单方法。本节介绍如何在安装了 Kubernetes 的内部（DGX-1）实例上安装 cnvrg。

### 前提条件

在完成安装之前，您必须在本地计算机上安装并准备以下依赖项：

- Kubectll
- Helm 3.x
- Kubernetes 集群 1.15 及更高版本

## 使用 Helm 部署

1. 要下载最新的 cnvrg Helm 图表，请运行以下命令：

```
helm repo add cnvrg https://helm.cnvrg.io
helm repo update
```

2. 在部署 cnvrg 之前，您需要集群的外部 IP 地址以及要部署 cnvrg 的节点的名称。要在内部 Kubernetes 集群上部署 cnvrg，请运行以下命令：

```
helm install cnvrg cnvrg/cnvrg --timeout 1500s --wait \ --set
global.external_ip=<ip_of_cluster> \ --set global.node=<name_of_node>
```

3. 运行 `helm install` 命令。所有服务和系统都会自动安装在集群上。此过程可能需要长达 15 分钟。
4. `helm install` 命令可能需要长达 10 分钟的时间。部署完成后，转到新部署的 cnvrg 的 URL 或将新集群添加为组织内部的资源。使用 `helm` 命令可向您告知正确的 URL。

```
Thank you for installing cnvrg.io!
Your installation of cnvrg.io is now available, and can be reached via:
Talk to our team via email at
```

5. 当所有容器的状态为 `running` 或 `complete` 时，已成功部署 cnvrg。它应类似于以下示例输出：

NAME	READY	STATUS	RESTARTS	AGE
cnvrg-app-69fbb9df98-6xrgf	1/1	Running	0	2m
cnvrg-sidekiq-b9d54d889-5x4fc	1/1	Running	0	2m
controller-65895b47d4-s96v6	1/1	Running	0	2m
init-app-vs-config-wv9c4	0/1	Completed	0	9m
init-gateway-vs-config-2zbpp	0/1	Completed	0	9m
init-minio-vs-config-cd2rg	0/1	Completed	0	9m
minio-0	1/1	Running	0	2m
postgres-0	1/1	Running	0	2m
redis-695c49c986-kcbt9	1/1	Running	0	2m
seeder-wh655	0/1	Completed	0	2m
speaker-5sghr	1/1	Running	0	2m

## 使用 ResNet50 和 Chest X 射线数据集的计算机视觉模型培训

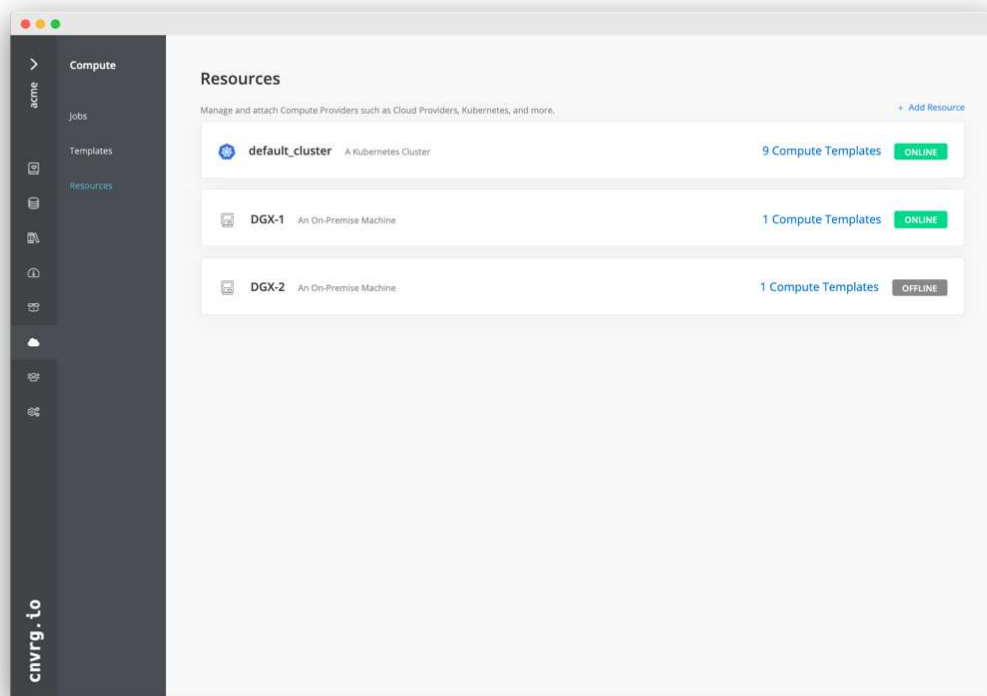
cnvrg.io AI 操作系统部署在 Kubernetes 设置中，并部署在由 NVIDIA DGX 系统提供支持的 NetApp ONTAP AI 架构上。为了进行验证，我们使用了 NIH Chest X ray 数据集，该数据集包含经去除身份识别的胸 x 射线图像。这些映像采用 PNG 格式。这些数据由 NIH 临床中心提供，可通过获取 ["NIH 下载站点"](#)。我们使用了一个 250 GB 的数据样本，其中包含 627 ， 615 个图像，分布在 15 个类别中。



数据集已上传到 cnvrg 平台，并在 NetApp AFF A800 存储系统的 NFS 导出中进行缓存。

## 设置计算资源

借助 cnvrg 架构和元数据计划功能，工程师和 IT 专业人员可以将不同的计算资源连接到一个平台。在我们的设置中，我们使用了为运行深度学习工作负载而部署的相同集群 cnvrg。如果需要连接其他集群，请使用 GUI，如以下屏幕截图所示。

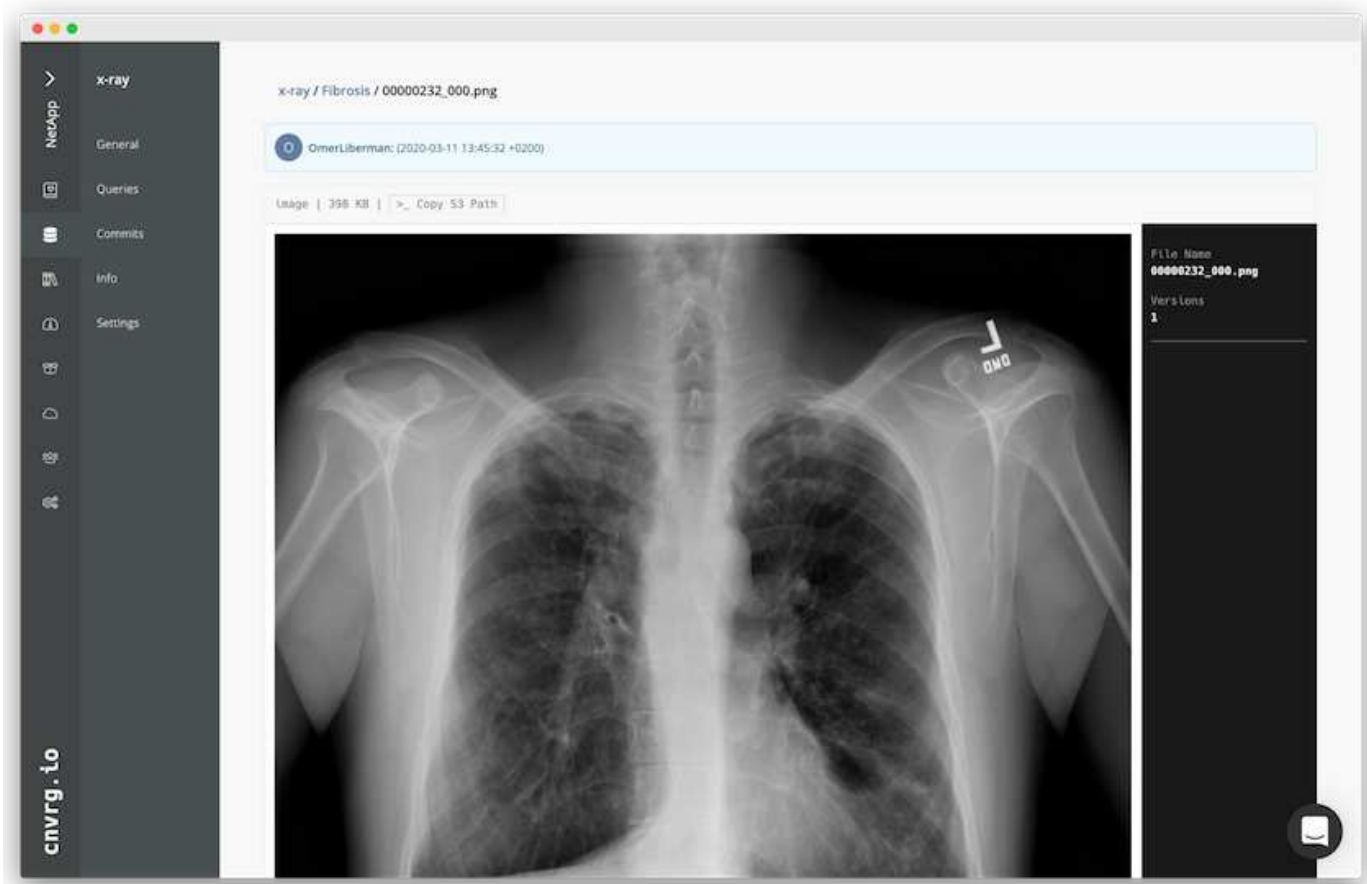


## 加载数据

要将数据上传到 cnvrg 平台，您可以使用 GUI 或 cnvrg CLI。对于大型数据集，NetApp 建议使用 CLI，因为它是一款强大，可扩展且可靠的工具，可以处理大量文件。

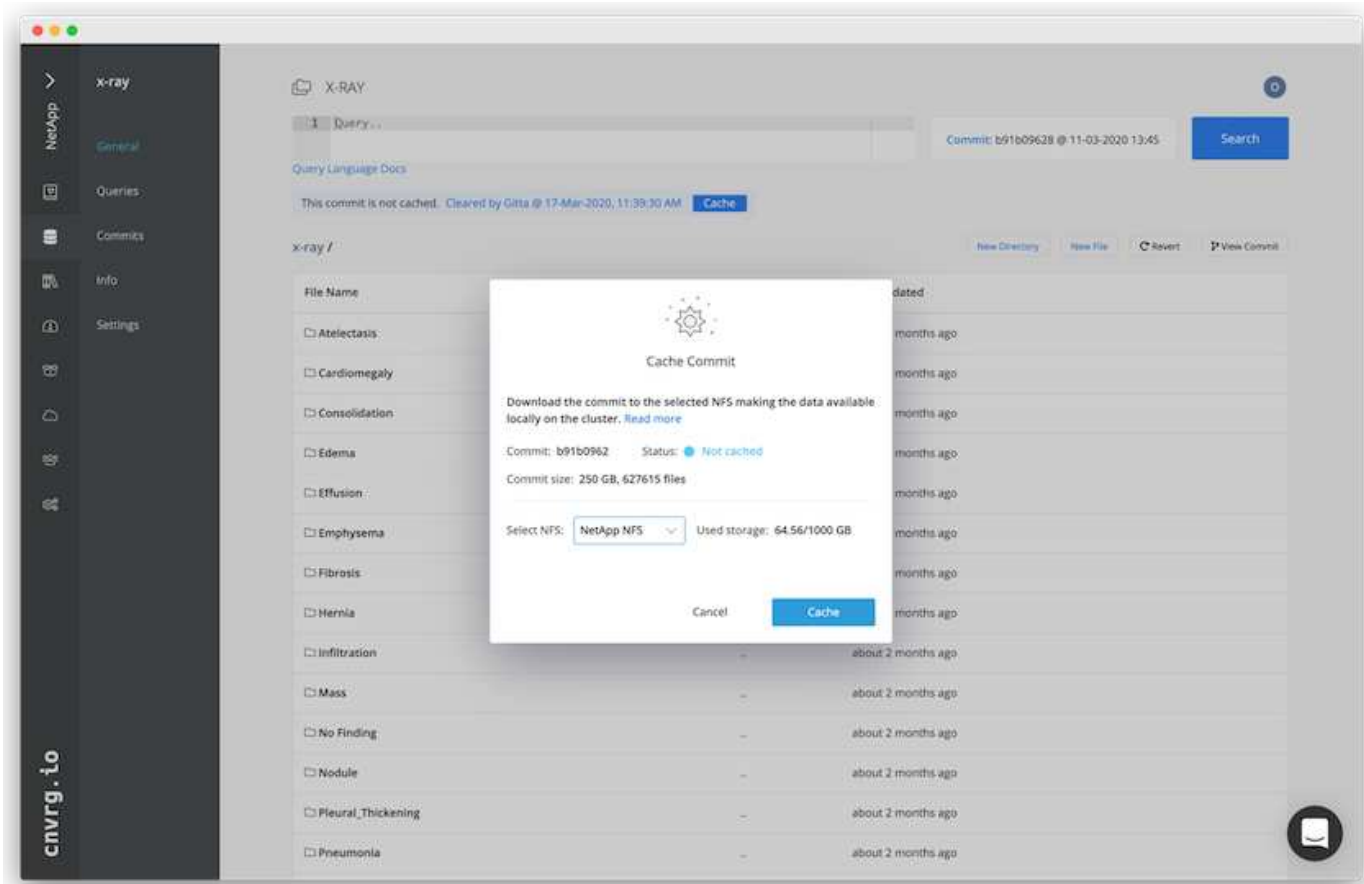
要上传数据，请完成以下步骤：

1. 下载 "[cnvrg 命令行界面](#)"。
2. 导航到 x-ray 目录。
3. 使用 `cnvrg data init` 命令初始化平台中的数据湖。
4. 使用 `cnvrg data sync` 命令将目录的所有内容上传到中央数据湖。将数据上传到中央对象存储（StorageGRID，S3 或其他）后，您可以使用 GUI 进行浏览。下图显示了一个已加载的胸 X 射线纤维化影像 PNG 文件。此外，cnvrg 会对数据进行版本控制，以便您构建的任何模型都可以复制到数据版本。



### Cach 数据

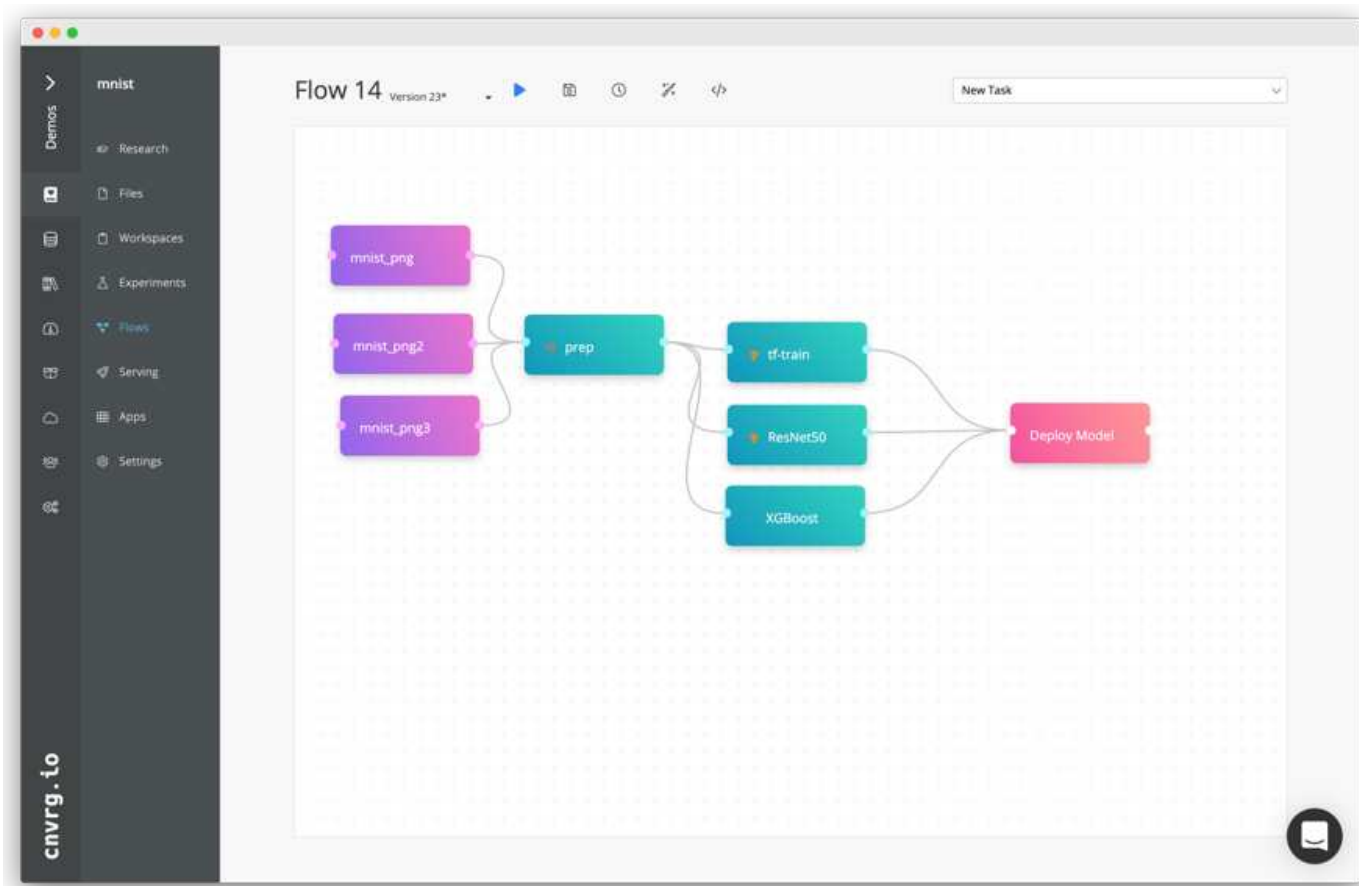
为了加快训练速度并避免为每个模型训练和实验下载 60 万多个文件，在数据最初上传到中央数据湖对象存储之后，我们使用了数据缓存功能。



用户单击缓存后，cnvrg 将从远程对象存储下载其特定提交中的数据，并将其缓存到 ONTAP NFS 卷上。完成后，可以使用这些数据进行即时培训。此外，如果数据在几天内未使用（例如用于模型训练或探索），则 cnvrg 会自动清除缓存。

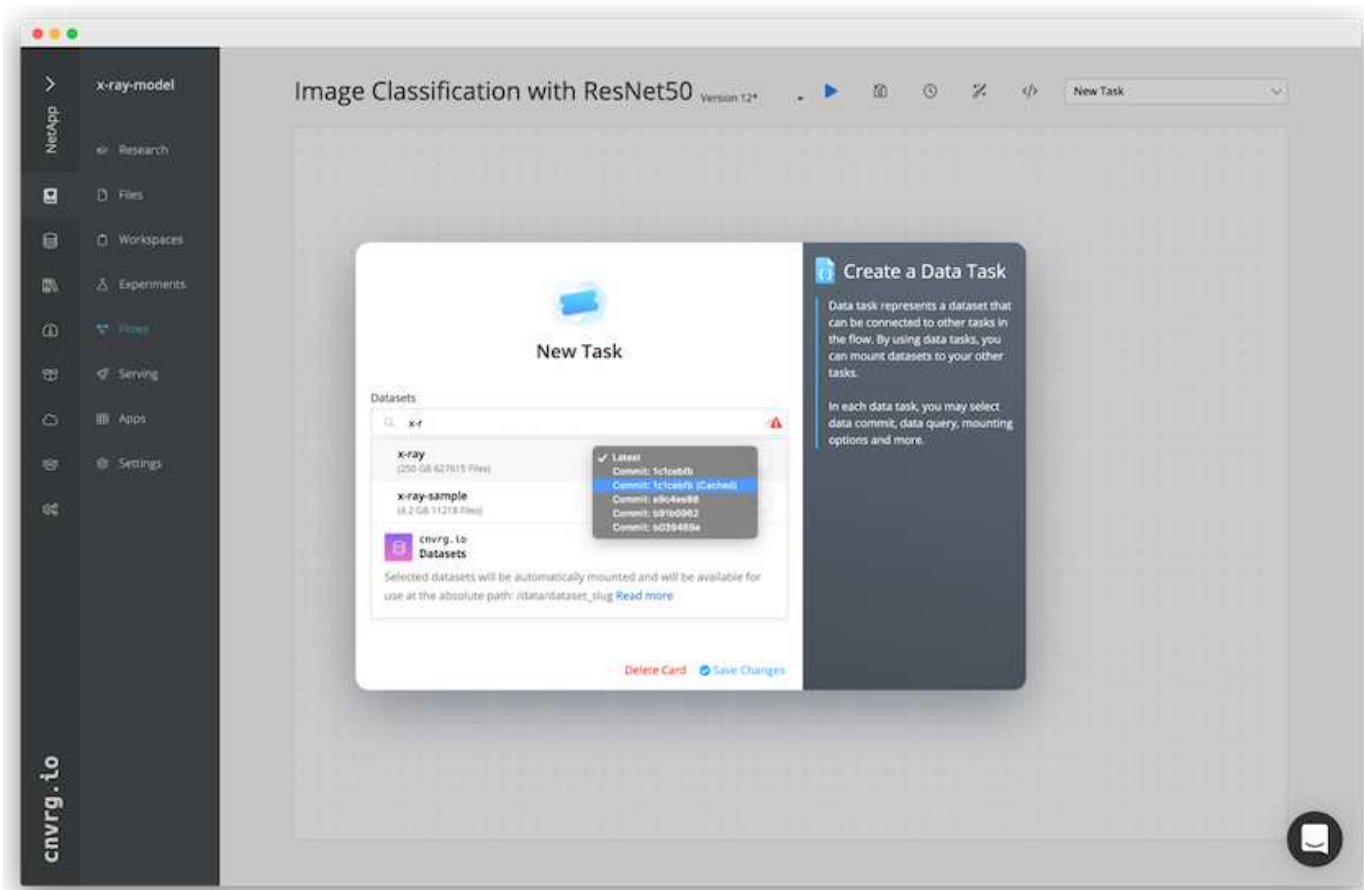
#### 使用缓存数据构建 ML 管道

借助 cnvrg 流，您可以轻松构建生产 ML 管道。流非常灵活，可用于任何类型的 ML 用例，并可通过 GUI 或代码创建。一个流中的每个组件都可以使用不同的 Docker 映像在不同的计算资源上运行，从而可以构建混合云和优化的 ML 管道。



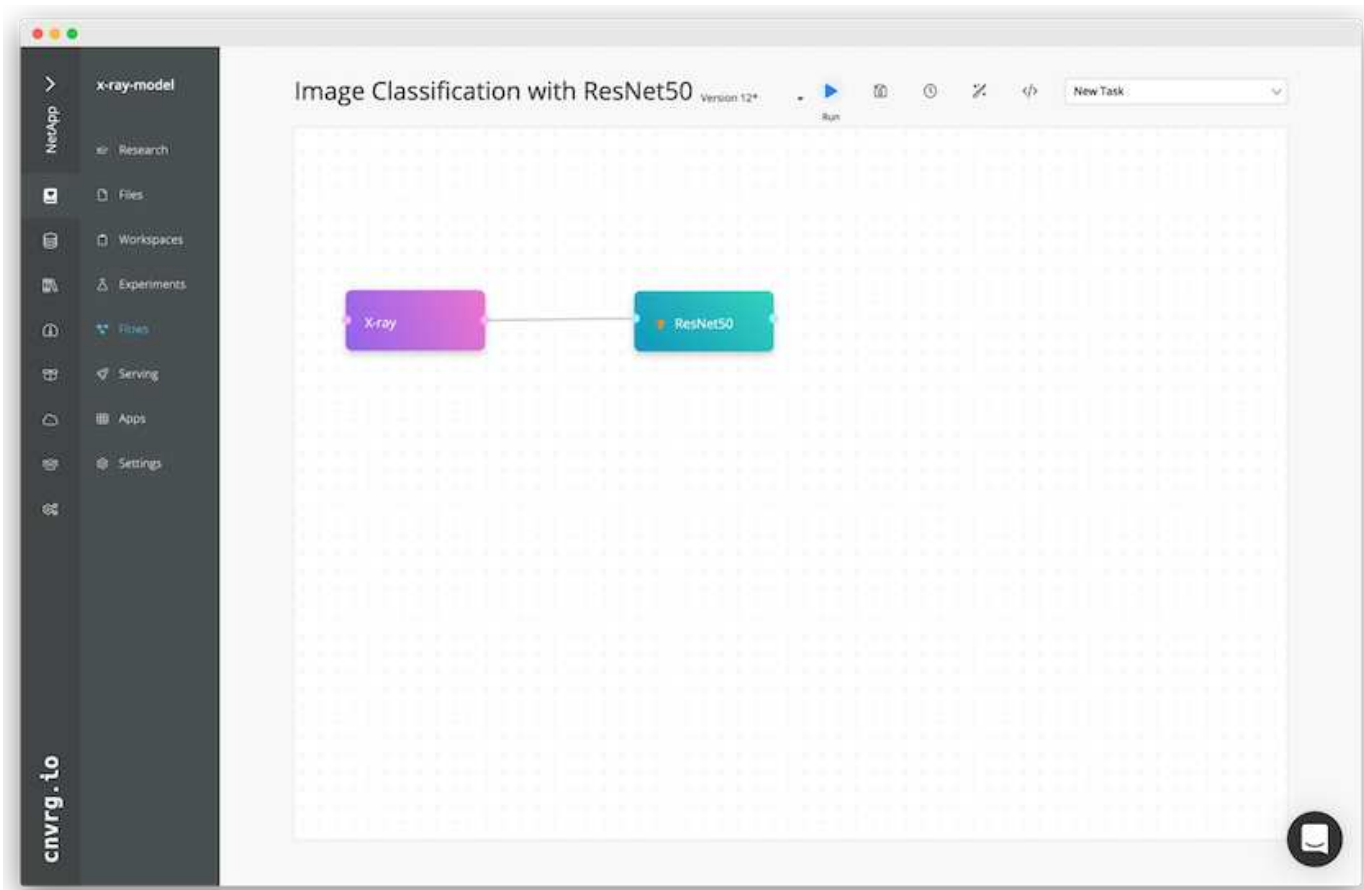
### 构建箱内 X 射线流：设置数据

我们已将数据集添加到新创建的流中。添加数据集时，您可以选择特定版本（提交）并指示是否需要缓存版本。在此示例中，我们选择了缓存的提交。



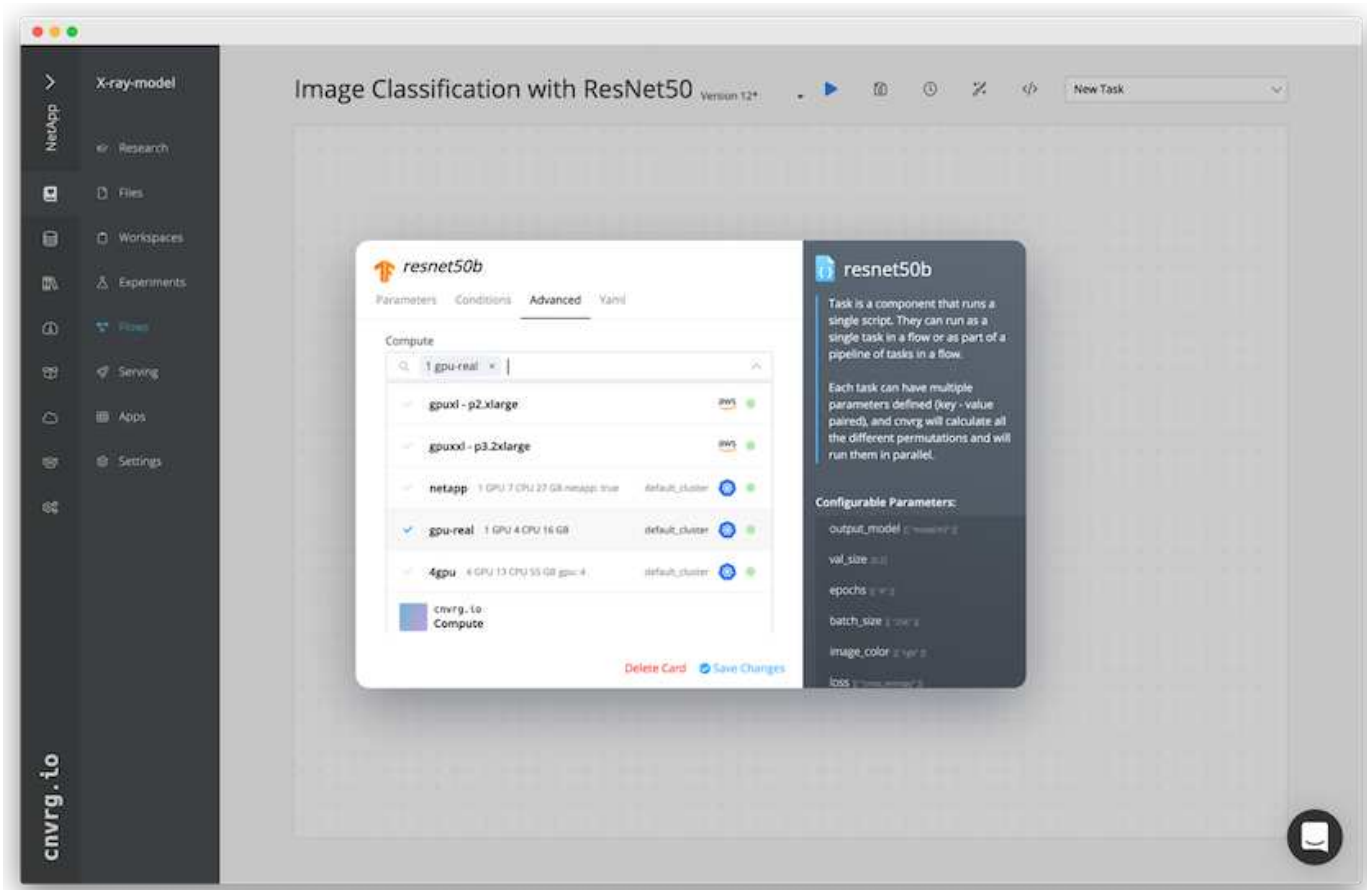
## 构建箱内 X 射线流：设置训练模型：ResNet50

在管道中，您可以添加所需的任何类型的自定义代码。在 cnvrg 中，还提供了 AI 库，这是一个可重复使用的 ML 组件集合。在 AI 库中，有算法，脚本，数据源以及其他解决方案可用于任何 ML 或深度学习流。在此示例中，我们选择了预构建的 ResNet50 模块。我们使用的是默认参数，例如 `batch_size : 128`，`epodchs : 10` 等。可以在 AI 库文档中查看这些参数。以下屏幕截图显示了将 X 线数据集连接到 ResNet50 的新流。



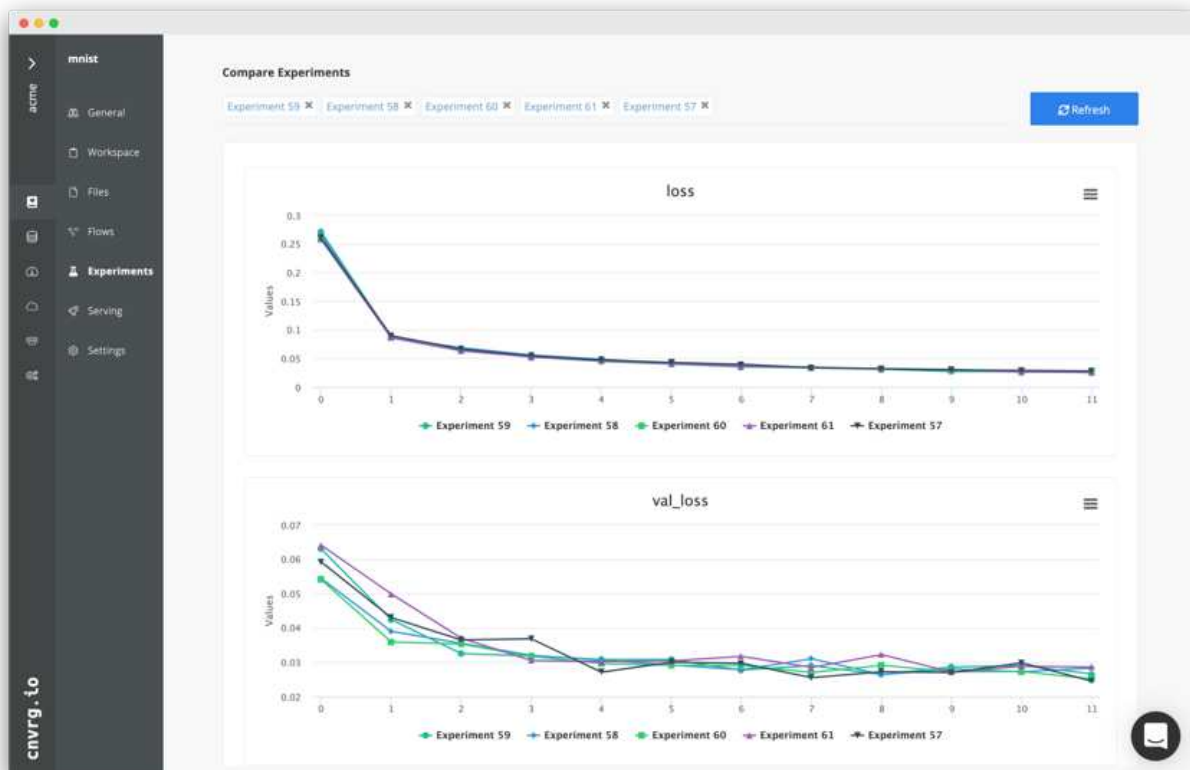
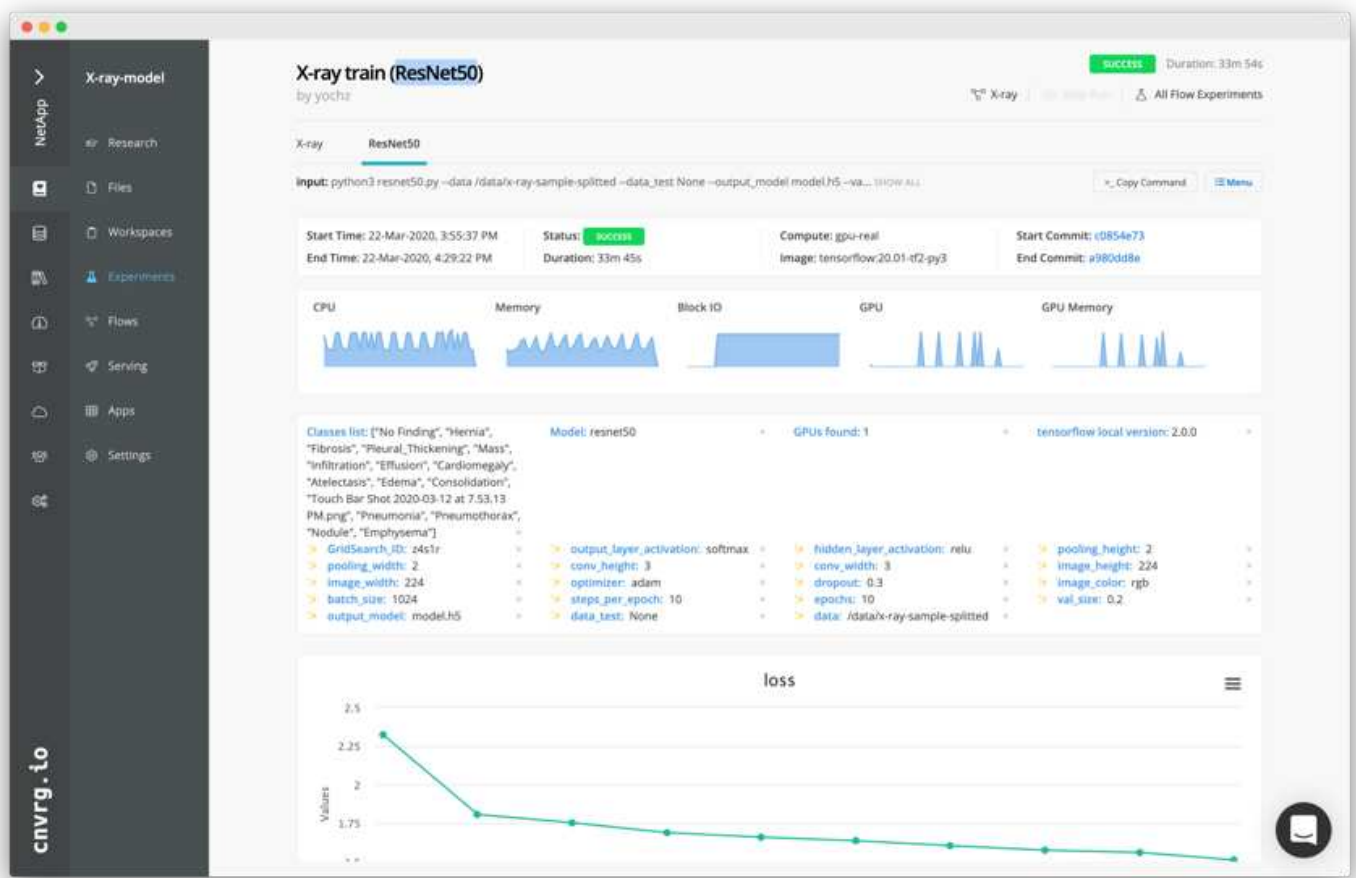
## 为 **ResNet50** 定义计算资源

cnvrg 流中的每个算法或组件都可以使用不同的 Docker 映像在不同的计算实例上运行。在我们的设置中，我们希望使用 NetApp ONTAP AI 架构在 NVIDIA DGX 系统上运行训练算法。在下图中，我们选择了 GPU Real，这是我们内部集群的计算模板和规范。我们还创建了一个模板队列并选择了多个模板。这样，如果无法分配 GPU-Real 资源（例如，如果其他数据科学家正在使用该资源），则可以通过添加云提供商模板来启用自动云突发功能。以下屏幕截图显示了如何使用 GPU Real 作为 ResNet50 的计算节点。



## 跟踪和监控结果

执行流量后，cnvrg 将触发跟踪和监控引擎。每次运行流程都会自动记录并实时更新。超参数，指标，资源使用情况（GPU 利用率等），代码版本，项目，日志，实验部分会自动提供，如以下两个屏幕截图所示。





## 结论

NetApp 和 cnvrg.io 合作为客户提供了一个完整的数据管理解决方案，用于 ML 和 DL 软件开发。ONTAP AI 可为任何规模的运营提供高性能计算和存储，而 cnvrg.io 软件可简化数据科学工作流并提高资源利用率。

## 致谢

- NetApp 技术营销工程师 Mike Oglesby
- NetApp 高级技术总监 Santosh Rao

## 从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请参见以下资源：

- cnvrg-io ( "<https://cnvrg.io>" ) :
  - cnvrg 核心 (免费 ML 平台)  
<https://cnvrg.io/platform/core>
  - Cnvrg 文档  
["https://app.cnvrg.io/docs"](https://app.cnvrg.io/docs)
- NVIDIA DGX-1 服务器：
  - 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 ( NGC )  
<https://www.nvidia.com/en-us/gpu-cloud/>
- NetApp AFF 系统：
  - AFF 产品规格  
<https://www.netapp.com/us/media/d-3582.pdf>
  - 适用于 AFF 的 NetApp FlashAdvantage  
<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/us/media/tr-4557.pdf>

- 适用于容器的 NetApp 持久存储:

- NetApp Trident

<https://netapp.io/persistent-storage-provisioner-for-kubernetes/>

- NetApp 互操作性表

- NetApp 互操作性表工具

<https://mysupport.netapp.com/matrix/#welcome>

- ONTAP AI 网络:

- Cisco Nexus 3232C 交换机

<https://www.cisco.com/c/en/us/products/switches/nexus-3232c-switch/index.html>

- Mellanox Spectrum 2000 系列交换机

[http://www.mellanox.com/page/products\\_dyn?product\\_family=251&mtag=sn2000](http://www.mellanox.com/page/products_dyn?product_family=251&mtag=sn2000)

- ML 框架和工具:

- DALI

<https://github.com/NVIDIA/DALI>

- TensorFlow : 适用于所有人的开源机器学习框架

<https://www.tensorflow.org/>

- Horovod : Uber 的 TensorFlow 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/>

- Docker

<https://docs.docker.com>

- Kubernetes

<https://kubernetes.io/docs/home/>

- NVIDIA DeepOps

<https://github.com/NVIDIA/deepops>

- Kubeflow

<http://www.kubeflow.org/>

- Jupyter 笔记本电脑服务器

<http://www.jupyter.org/>

- 数据集和基准测试：

- NIH Chest X 射线数据集

<https://nihcc.app.box.com/v/ChestXray-NIHCC>

- 王晓松, 彭义文, 卢乐, 陆志勇, 莫哈马达迪·巴格里, Ronald Summers, ChestX-ray8: 《医院级胸 X 射线数据库和常见胸病的弱监督分类和本地化基准》, IEEE CVPR, 第页3462-3471, 2017TR-4841-0620

## TR-4732：大数据分析数据到人工智能

NetApp 公司 Karthikeyan Nagalingam

本文档介绍如何将大数据分析数据和 HPC 数据迁移到 AI。AI 通过 NFS 导出处理 NFS 数据，而客户通常将其 AI 数据存储在 HDFS，Blob 或 S3 存储等大数据分析平台以及 GPFS 等 HPC 平台中。本白皮书提供了使用 NetApp XCP 和 NIPAM 将大数据分析数据和 HPC 数据迁移到 AI 的准则。我们还讨论了将数据从大数据和 HPC 迁移到 AI 的业务优势。

### 概念和组件

#### 大数据分析存储

大数据分析是 HDFS 的主要存储提供商。客户通常使用与 Hadoop 兼容的文件系统（HCFS），例如 Windows Azure Blob Storage，MapR 文件系统（MapR-FS）和 S3 对象存储。

#### 常规并行文件系统

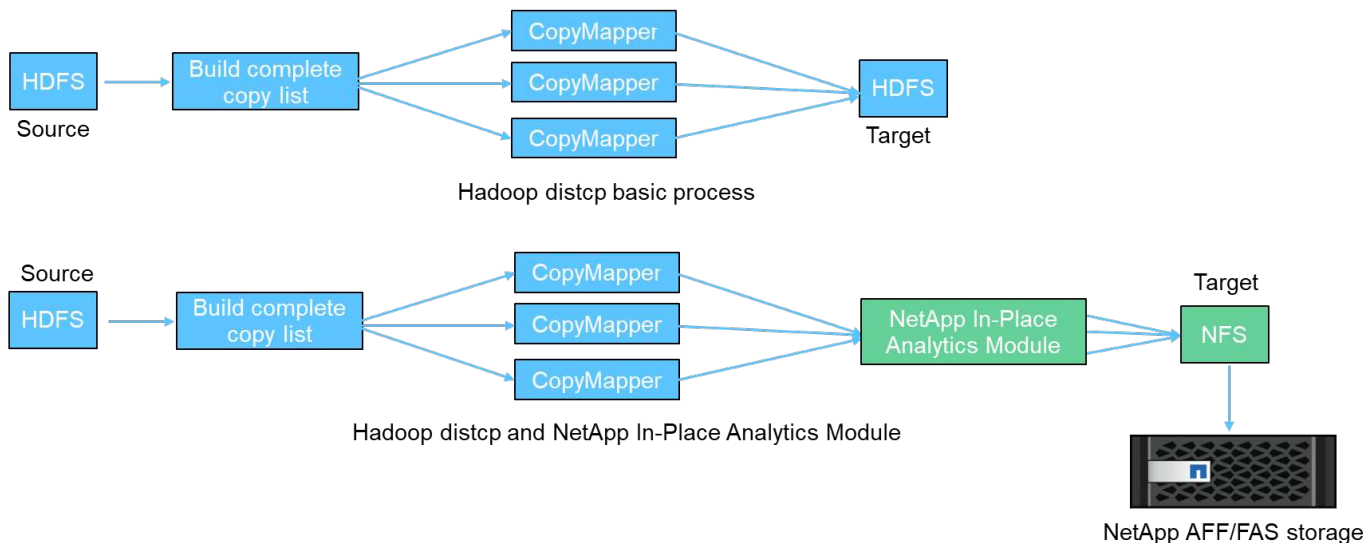
IBM 的 GPFS 是一种企业级文件系统，可替代 HDFS。利用 GPF，应用程序可以灵活地确定块大小和复制布局，从而提供良好的性能和效率。

#### NetApp 原位分析模块

NetApp 原位分析模块（NIPAM）可作为 Hadoop 集群访问 NFS 数据的驱动程序。它包含四个组件：连接池，NFS InputStream，文件句柄缓存和 NFS OutputStream。有关详细信息，请参见 ["TR-4382：NetApp 原位分析模块。"](#)

#### Hadoop 分布式副本

Hadoop 分布式副本（DistCp）是一种分布式副本工具，用于执行大型集群间和集群内应对任务。此工具使用 MapReduce 进行数据分发，错误处理和报告。它会扩展文件和目录列表，并输入这些文件和目录以映射任务，从而从源列表复制数据。下图显示了 HDFS 和非 HDFS 中的 DistCp 操作。



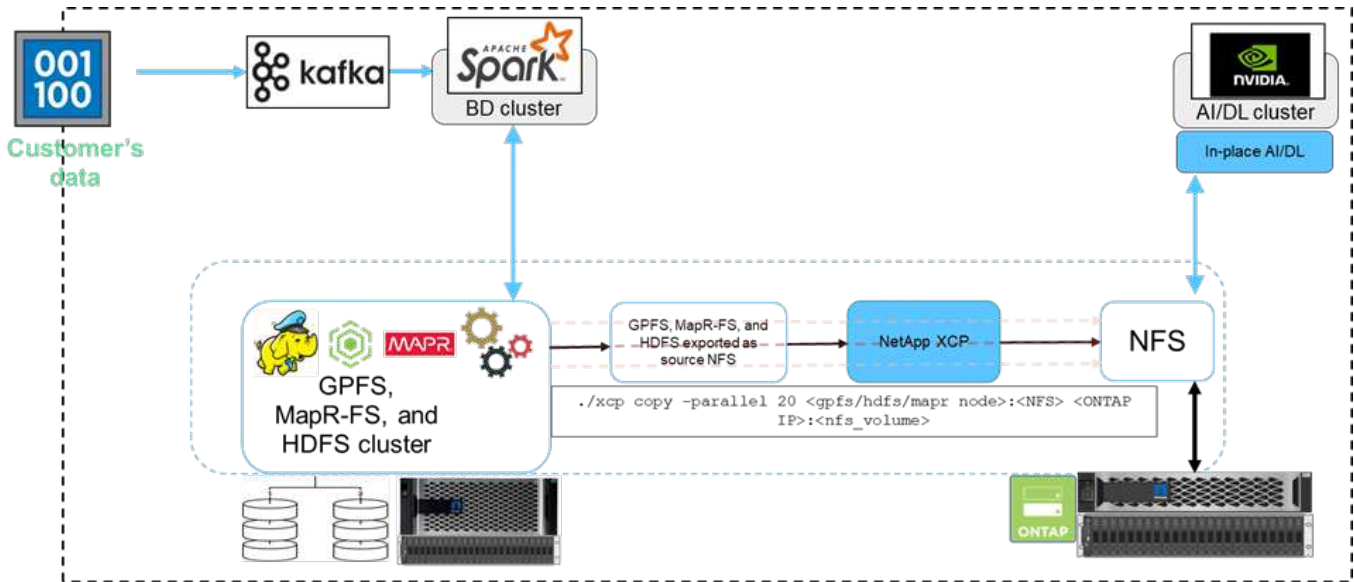
Hadoop DistCp 可在两个 HDFS 系统之间移动数据，而无需使用其他驱动程序。NetApp 为非 HDFS 系统提供了驱动程序。对于 NFS 目标，NIPAM 提供了用于复制数据的驱动程序，Hadoop DistCp 在复制数据时使用这些数据与 NFS 目标进行通信。

### NetApp Cloud Volumes Service

NetApp Cloud Volumes Service 是一种性能极高的云原生文件服务。此服务可通过快速启动和关闭资源并使用 NetApp 功能来提高工作效率并减少员工停机时间，帮助客户加快产品上市速度。Cloud Volumes Service 是灾难恢复和备份到云的理想替代方案，因为它可以减少数据中心的整体占用空间，并减少原生公有云存储的使用量。

### NetApp XCP

NetApp XCP 是一款客户端软件，可实现快速可靠的任意到 NetApp 和 NetApp 到 NetApp 数据迁移。此工具用于将大量非结构化 NAS 数据从任何 NAS 系统复制到 NetApp 存储控制器。XCP 迁移工具使用多核多通道 I/O 流式引擎，可以并行处理多个请求，例如数据迁移，文件或目录列表以及空间报告。这是默认的 NetApp 数据迁移工具。您可以使用 XCP 将数据从 Hadoop 集群和 HPC 复制到 NetApp NFS 存储。下图显示了使用 XCP 从 Hadoop 和 HPC 集群到 NetApp NFS 卷的数据传输。



## NetApp BlueXP复制和同步

NetApp BlueXP复制和同步是一种混合数据复制软件即服务、可在内部存储和云存储之间安全无缝地传输和同步NFS、S3和CIFS数据。此软件用于数据迁移，归档，协作，分析等。传输数据后、BlueXP复制和同步功能会持续同步源和目标之间的数据。接下来，它会传输增量数据。它还可以保护您自己网络，云或内部环境中的数据。此软件采用按需购买模式，可提供经济高效的解决方案并为数据传输提供监控和报告功能。

## 边缘人工智能推理—采用联想 ThinkSystem 的 NetApp —解决方案设计

### TR-4886：《前沿人工智能推理—采用联想 ThinkSystem 的 NetApp —解决方案设计》

NetApp 公司 Sathish Thyagarajan ， 联想公司 Miroslav Hodak

本文档介绍了一种计算和存储架构，用于在边缘环境中针对新兴应用程序场景在 NetApp 存储控制器和联想 ThinkSystem 服务器上部署基于 GPU 的人工智能（AI）推理。

#### 摘要

高级驾驶辅助系统（ADAS），行业 4.0，智能城市和物联网（IoT）等多种新兴应用场景需要在近乎零的延迟下处理持续数据流。本文档介绍了一种计算和存储架构，用于在边缘环境中为 NetApp 存储控制器和联想 ThinkSystem 服务器部署基于 GPU 的人工智能（AI）推理。本文档还提供了行业标准 MLPerf 推理基准测试的性能数据，用于评估配备 NVIDIA T4 GPU 的边缘服务器上的各种推理任务。我们对脱机，单流和多流推理情形的性能进行了调查，并显示具有经济高效的共享网络存储系统的架构具有高性能，为多个边缘服务器的数据和模型管理提供了一个中心。

#### 简介

企业越来越多地在网络边缘生成海量数据。为了从智能传感器和物联网数据中获得最大价值，企业正在寻找支持边缘计算的实时事件流式解决方案。因此，在数据中心以外的边缘执行计算要求苛刻的作业的情况越来越多。AI 推理是这一趋势的推动因素之一。边缘服务器可以为这些工作负载提供足够的计算能力，尤其是在使用加速器时，但有限的存储通常是问题描述，尤其是在多服务器环境中。在本文档中，我们将介绍如何在边缘环境中部署共享存储系统，以及该系统如何在不影响性能的情况下为 AI 推理工作负载带来优势。

本文档介绍了边缘 AI 推理的参考架构。它将多个联想 ThinkSystem 边缘服务器与一个 NetApp 存储系统相结合，创建了一个易于部署和管理的解决方案。本指南旨在为各种情形下的实际部署提供一个基线指南，例如，在工厂车间安装多个摄像头和工业传感器，零售交易中的销售点（POS）系统或用于识别自动驾驶车辆中的视觉异常的完全自驾（FSD）系统。

本文档介绍了对由联想 ThinkSystem SE350 边缘服务器和入门级 NetApp AFF 和 EF 系列存储系统组成的计算和存储配置的测试和验证。这些参考架构可为 AI 部署提供高效且经济高效的解决方案，同时还可通过 NetApp ONTAP 和 NetApp SANtricity 数据管理软件提供全面的数据服务，集成数据保护，无缝可扩展性以及云连接数据存储。

#### 目标受众

本文档面向以下受众：

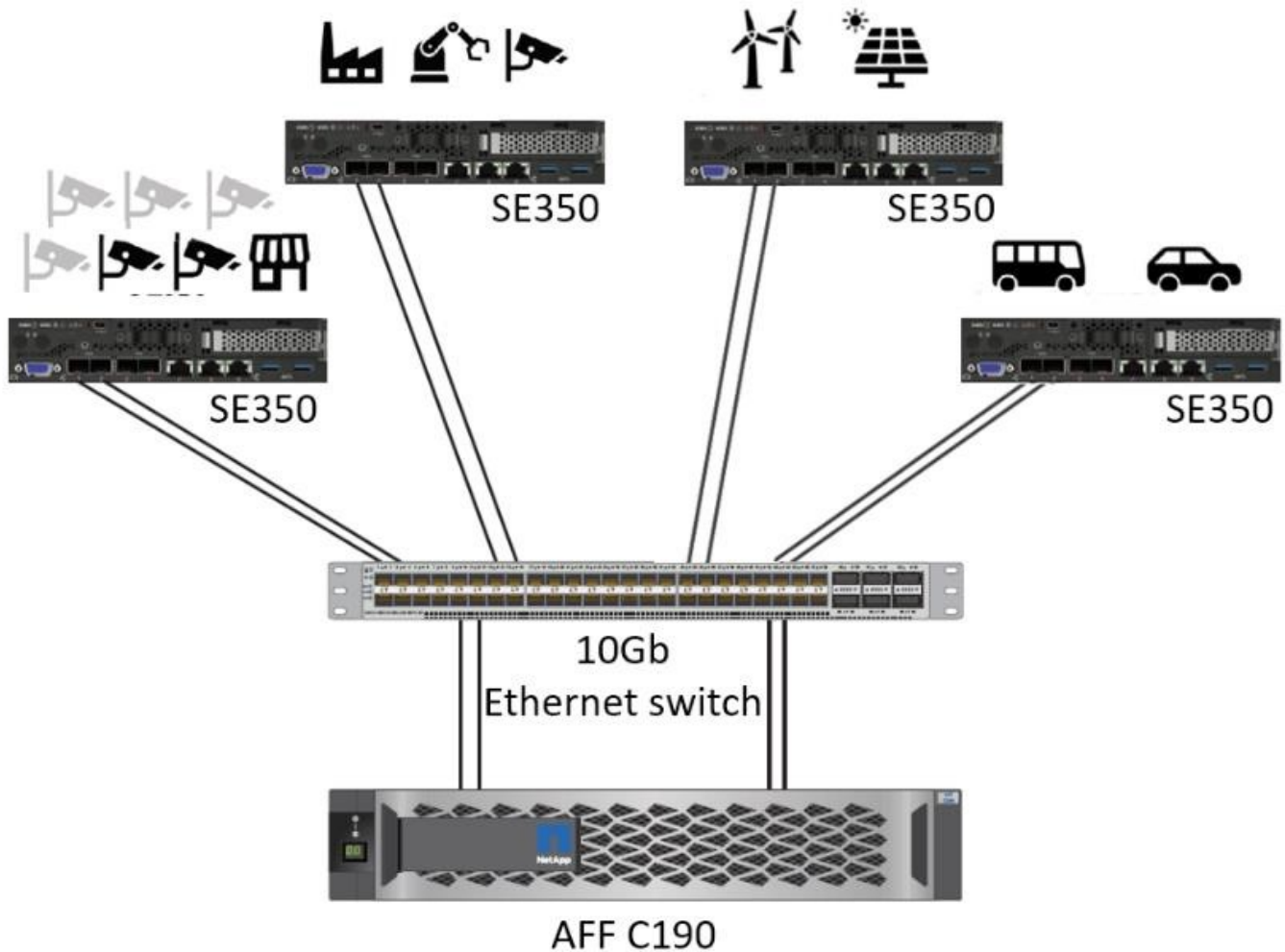
- 希望在边缘将 AI 产品化的业务主管和企业架构师。
- 数据科学家，数据工程师，人工智能 / 机器学习（AI / 机器学习，ML）研究人员和人工智能系统开发人

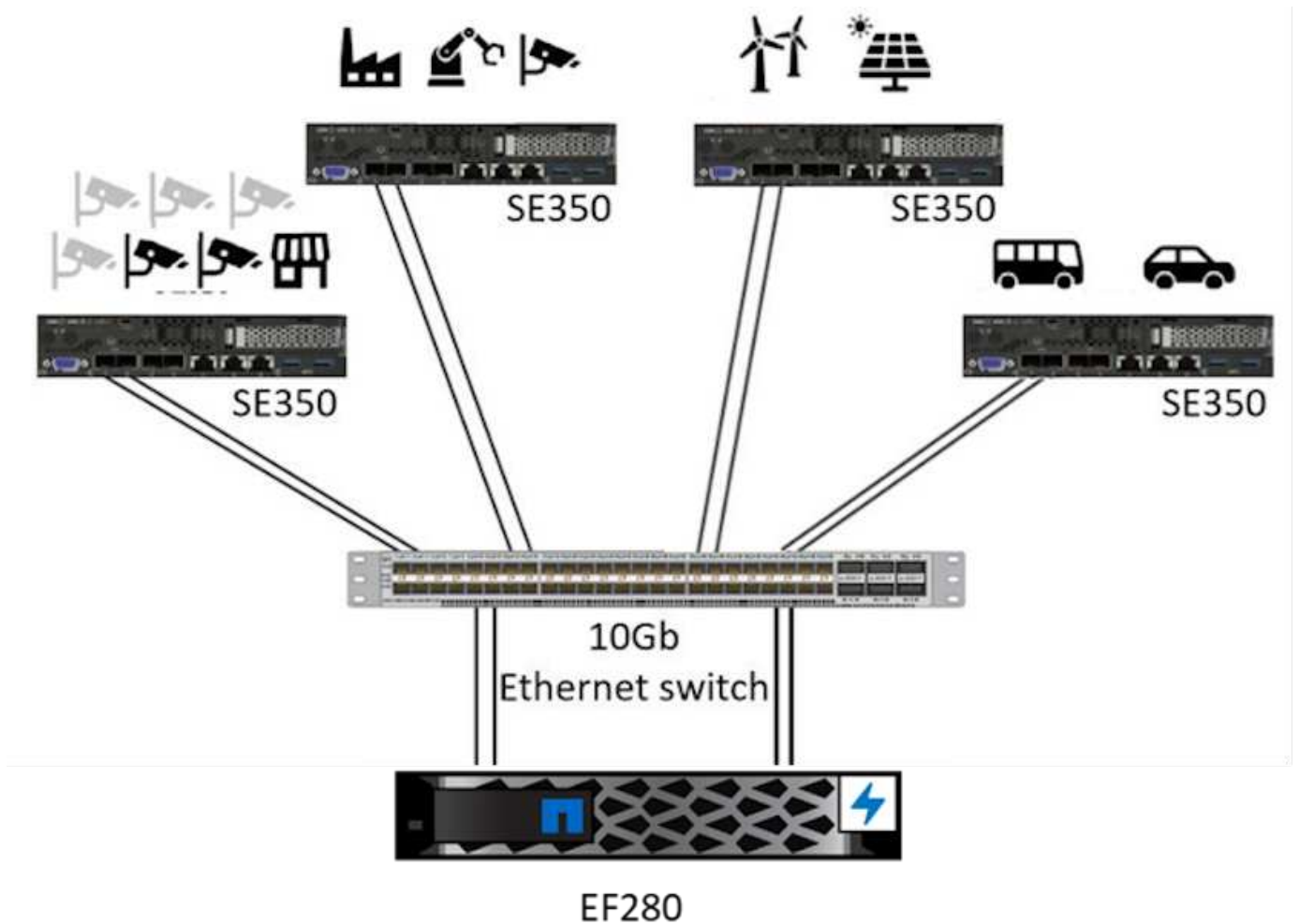
员。

- 设计用于开发 AI/ML 模型和应用程序的解决方案的企业架构师。
- 数据科学家和 AI 工程师正在寻找部署深度学习（DL）和 ML 模型的高效方法。
- 边缘设备管理器和边缘服务器管理员，负责部署和管理边缘推理模型。

#### 解决方案架构

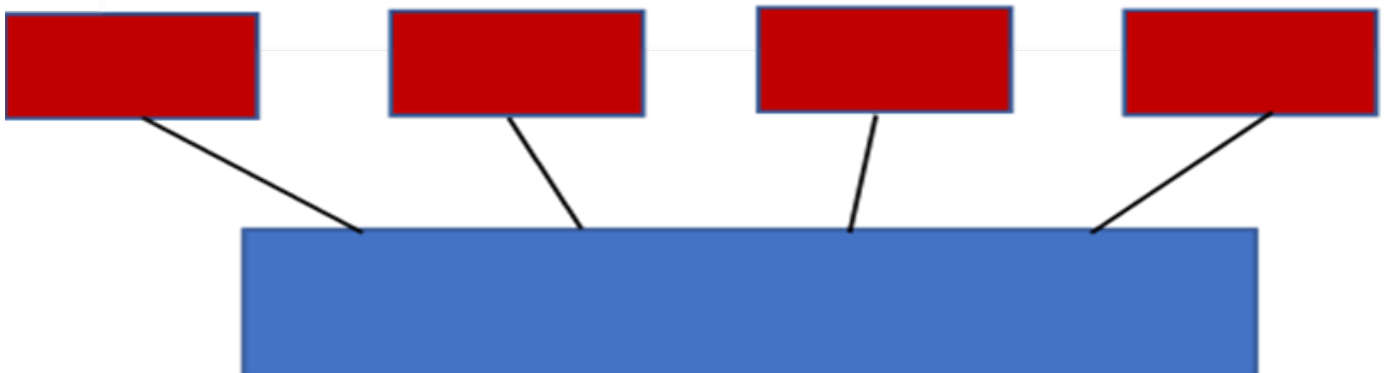
此联想 ThinkSystem 服务器和 NetApp ONTAP 或 NetApp SANtricity 存储解决方案旨在利用 GPU 与传统 CPU 的处理能力处理大型数据集上的 AI 推理。此验证通过一个架构来展示高性能和最佳数据管理，该架构使用一个或多个与单个 NetApp AFF 存储系统互连的联想 SR350 边缘服务器，如下两个图所示。





下图中的逻辑架构概述显示了此架构中计算和存储元素的角色。具体而言，它显示以下内容：

- 边缘计算设备对从摄像机，传感器等接收的数据执行推理。
- 一种可用于多种用途的共享存储元素：
  - 为推理模型和执行推理所需的其他数据提供一个中央位置。计算服务器可以直接访问存储，并在网络中使用推理模型，而无需将其复制到本地。
  - 此处推送更新的型号。
  - 对边缘服务器接收的输入数据进行归档，以供日后分析。例如，如果边缘设备连接到摄像机，则存储元素会保留摄像机捕获的视频。





红色	蓝色
联想计算系统	NetApp AFF 存储系统
边缘设备对来自摄像机，传感器等的输入执行推理。	共享存储，用于存放边缘设备的推理模型和数据，以供日后分析。

NetApp 和联想解决方案具有以下主要优势：

- GPU 加快了边缘计算速度。
- 部署从共享存储提供支持和管理的多个边缘服务器。
- 强大的数据保护功能，可满足低恢复点目标（RPO）和恢复时间目标（RTO）的要求，而不会丢失任何数据。
- 利用 NetApp Snapshot 副本和克隆优化数据管理，以简化开发工作流。

如何使用此架构

本文档将验证建议架构的设计和性能。但是，我们尚未测试某些软件级别的组件，例如容器，工作负载或模型管理以及与内部云或数据中心的数据同步，因为它们是特定于部署情形的。此处有多种选择。

在容器管理级别，Kubernetes 容器管理是一个不错的选择，完全上游版本（Canonical）或适用于企业部署的修改版本（Red Hat）均支持此功能。。"[NetApp AI 控制平台](#)" 使用 NetApp Trident 和新添加的 "[NetApp DataOps 工具包](#)" 为数据科学家和数据工程师提供内置可追溯性，数据管理功能，接口和工具，以便与 NetApp 存储集成。Kubeflow 是适用于 Kubernetes 的 ML 工具包，可在 TensorFlow Serving 或 NVIDIA Triton 推理服务器等多个平台上提供额外的 AI 功能以及对型号版本控制和 KFServing 的支持。另一个选项是 NVIDIA EGX 平台，它可提供工作负载管理以及对支持 GPU 的 AI 推理容器目录的访问。但是，这些选项可能需要大量的精力和专业知识才能投入生产，并且可能需要第三方独立软件供应商（ISV）或顾问的协助。

解决方案区域

AI 推理和边缘计算的主要优势是设备能够在无延迟的情况下以高质量计算，处理和分析数据。本文档中需要介绍的边缘计算用例太多，但下面是几个突出的示例：

汽车：自动驾驶汽车

典型的边缘计算图示位于自动驾驶汽车（AV）中的高级驾驶辅助系统（ADAS）中。无人驾驶汽车中的 AI 必须快速处理来自摄像机和传感器的大量数据，才能成为成功的安全驱动器。在对象和人类之间进行解释所花费的时间过长可能意味着生命或死亡，因此能够尽可能接近车辆处理数据至关重要。在这种情况下，一个或多个边缘计算服务器处理来自摄像机，雷达，LIDAR 和其他传感器的输入，而共享存储则保存推理模型并存储来自传感器的输入数据。

医疗保健：患者监控

AI 和边缘计算的最大影响之一是，它能够在家庭护理和集中护理部门（ICU）中增强对慢性病患者的持续监控。监控 Insulin 级别，呼吸，神经活动，心率和消化系统功能的边缘设备中的数据需要即时分析数据，必须立即对这些数据执行操作，因为拯救生命的行动时间有限。

零售：无收银员付款

边缘计算可以为 AI 和 ML 提供支持，帮助零售商缩短结账时间并增加流量。无收银员系统支持各种组件，例如：



- 身份验证和访问。将物理购物者连接到经过验证的帐户并允许访问零售空间。
- 清单监控。使用传感器，RFID 标签和计算机视觉系统帮助确认买家选择或取消选择商品。

此处，每个边缘服务器都会处理每个签出计数器，而共享存储系统则充当一个中央同步点。

#### 金融服务：信息亭的人员安全和防止欺诈

银行组织正在使用 AI 和边缘计算来创新和打造个性化的银行体验。利用实时数据分析和人工智能推理的交互式信息亭现在不仅可以帮助客户提取资金，还可以通过从摄像机捕获的图像主动监控信息亭，以识别对人类安全或欺诈行为的风险。在这种情况下，边缘计算服务器和共享存储系统连接到交互式信息亭和摄像机，以帮助银行使用 AI 推理模型收集和处理数据。

#### 制造业：行业 4.0

第四次工业革命（工业 4.0）已经开始，同时也出现了 Smart Factory 和 3D 打印等新兴趋势。为迎接数据主导的未来，大规模机器到机器（M2M）通信和物联网已集成在一起，可提高自动化程度，无需人工干预。制造业已经高度自动化，增加 AI 功能是长期趋势的自然延续。AI 可实现自动化操作，借助计算机视觉和其他 AI 功能，可以实现自动化操作。您可以自动执行质量控制或依赖于人类远见或决策的任务，以便更快地分析工厂车间内装配线上的材料，从而帮助制造工厂满足所需的 ISO 安全和质量管理标准。此处，每个计算边缘服务器都连接到一组传感器，用于监控制造过程，更新的推理模型会根据需要推送到共享存储。

#### 电信：防锈检测，塔式检查和网络优化

电信行业使用计算机视觉和人工智能技术来处理图像，这些图像可自动检测到是否存在防腐问题，并识别含有防腐问题的单元塔，因此需要进一步检查。近年来，使用无人机图像和 AI 模型来识别塔中不同的区域，以分析防腐，表面裂纹和防腐的情况有所增加。对 AI 技术的需求继续增长，这些技术可以高效地检查电信基础架构和单元塔，定期评估其降级情况，并在需要时及时修复。

此外，电信领域另一个新兴的使用情形是，使用 AI 和 ML 算法预测数据流量模式，检测支持 5G 的设备，以及自动化和增强多输入和多输出（MIMO）能源管理。在无线电塔上使用了 MIMO 硬件来增加网络容量，但这会带来额外的能源成本。在单元站点上部署的 ML 型号的 "MIMO 休眠模式" 可以预测是否高效使用了无线电，并有助于降低移动网络运营商（MNO）的能耗成本。AI 推理和边缘计算解决方案可帮助 MNO 减少来回传输到数据中心的数据量，降低 TCO，优化网络运营并提高最终用户的整体性能。

## 技术概述

本节介绍此 AI 解决方案的技术基础。

### NetApp AFF 系统

一流的 NetApp AFF 存储系统支持在边缘进行 AI 推理部署，通过行业领先的性能，卓越的灵活性，云集成和一流的数据管理满足企业级存储需求。NetApp AFF 系统专为闪存设计，可帮助加速，管理和保护业务关键型数据。

- 入门级 NetApp AFF 存储系统基于 FAS2750 硬件和 SSD 闪存介质
- HA 配置中的两个控制器



NetApp 入门级 AFF C190 存储系统支持以下功能：

- 最大驱动器数为 24 个 960 GB SSD
- 两种可能的配置：
  - 以太网（10GbE）：4 个 10GBASE-T（RJ-45）端口
  - 统一（16 Gb FC 或 10GbE）：4 个统一目标适配器 2（UTA2）端口
- 最大有效容量为 50.5 TB



对于 NAS 工作负载，一个入门级 AFF C190 系统支持连续读取吞吐量为 4.4 GBps，延迟为 1 毫秒或更短的小型随机读取吞吐量为 230 K IOPS。

### NetApp AFF A220

NetApp 还提供了其他入门级存储系统，可为大规模部署提供更高的性能和可扩展性。对于 NAS 工作负载，一个入门级 AFF A220 系统支持：

- 顺序读取的吞吐量为 6.2 GBps
- 在延迟不超过 1 毫秒的情况下进行小型随机读取时，可实现 375000 IOPS
- 最大驱动器数为 144 个 960 GB，3.8 TB 或 7.6 TB SSD
- AFF A220 可扩展到 1 PB 以上的有效容量

### NetApp AFF A250

- 最大有效容量为 35 PB，最大横向扩展为 2-24 个节点（12 个 HA 对）
- 与 AFF A220 相比，性能提高≥ 45%
- 440,000 次 IOPS 随机读取 @1 毫秒
- 基于最新的 NetApp ONTAP 版本 ONTAP 9.8 构建
- 利用两个 25 Gb 以太网实现 HA 和集群互连

## NetApp E 系列 EF 系统

EF 系列是一组入门级和中端全闪存 SAN 存储阵列，可通过 NetApp SANtricity 软件加快数据访问速度，并帮助您更快地从中获得价值。这些系统提供 SAS 和 NVMe 闪存存储，并为您提供经济实惠的至极高 IOPS，100 微秒以下的响应时间以及高达 44 GBps 的带宽，使其成为混合工作负载以及 AI 推理和高性能计算（HPC）等要求苛刻的应用程序的理想之选。

下图显示了 NetApp EF280 存储系统。



### NetApp EF280

- 支持 32 Gb/16 Gb FC，25 Gb/10 Gb iSCSI 和 12 Gb SAS
- 最大有效容量为 96 个驱动器，总容量为 1.5 PB
- 10 Gbps 吞吐量（顺序读取）
- 30 万次 IOPS（随机读取）
- NetApp EF280 是 NetApp 产品组合中成本最低的全闪存阵列（AFA）

### NetApp EF300

- 24 个 NVMe SSD 驱动器，总容量为 367 TB
- 扩展选项，总容量为 240 个 NL-SAS HDD，96 个 SAS SSD 或两者的组合
- 100 GB NVMe/IB，NVMe/RoCE，iSE/IB 和 SRP/IB
- 32 GB NVMe/FC，FCP
- 25 GB iSCSI
- 20 Gbps（顺序读取）
- 670K IOPS（随机读取）



有关详细信息，请参见 ["NetApp EF 系列 NetApp EF 系列全闪存阵列 EF600，F300，EF570 和 EF280 产品规格"](#)。

## NetApp ONTAP 9.

ONTAP 9.8.1 是 NetApp 推出的最新一代存储管理软件，可帮助企业打造现代化的基础架构并过渡到云就绪数据

中心。借助行业领先的数据管理功能，无论数据位于何处，ONTAP 都可以通过一组工具来管理和保护数据。您还可以将数据自由移动到需要的任何位置：边缘，核心或云。ONTAP 9.8.1 提供了许多功能，可简化数据管理，加快和保护关键数据，并在混合云架构中实现下一代基础架构功能。

#### 简化数据管理

数据管理对于企业 IT 运营至关重要，以便将适当的资源用于应用程序和数据集。ONTAP 具有以下功能，可简化操作并降低总运营成本：

- \* 实时数据缩减和扩展的重复数据删除。\* 数据缩减可减少存储块中浪费的空间，重复数据删除可显著提高有效容量。此适用场景数据存储在本地，并分层到云。
- \* 最低，最高和自适应服务质量（AQoS）。\* 细粒度服务质量（QoS）控制有助于在高度共享的环境中保持关键应用程序的性能水平。
- \* NetApp FabricPool。\* 此功能可将冷数据自动分层到公有和私有云存储选项，包括 Amazon Web Services（AWS），Azure 和 NetApp StorageGRID Storage 解决方案。有关 FabricPool 的详细信息，请参见 ["TR-4598"](#)。

#### 加速和保护数据

ONTAP 9 可提供卓越的性能和数据保护，并通过以下方式扩展这些功能：

- \* 性能和低延迟。\* ONTAP 可提供尽可能高的吞吐量，并尽可能降低延迟。
- \* 数据保护。\* ONTAP 提供内置数据保护功能，并在所有平台之间进行通用管理。
- \* NetApp 卷加密（NVE）。\* ONTAP 提供原生卷级加密，并支持板载和外部密钥管理。
- \* 多租户和多因素身份验证。\* ONTAP 支持以最高的安全性级别共享基础架构资源。

#### Future-Proof 基础架构

ONTAP 9 具有以下功能，可满足不断变化的苛刻业务需求：

- \* 无缝扩展和无中断运行。\* ONTAP 支持向现有控制器和横向扩展集群无中断添加容量。客户可以升级到 NVMe 和 32 Gb FC 等最新技术，而无需进行成本高昂的数据迁移或中断。
- \* 云连接。\* ONTAP 是云连接最广泛的存储管理软件，可在所有公有云中选择软件定义的存储（ONTAP Select）和云原生实例（NetApp Cloud Volumes Service）。
- \* 与新兴应用程序集成。\* ONTAP 使用支持现有企业级应用程序的相同基础架构，为下一代平台和应用程序（例如自动驾驶汽车，智能城市和行业 4.0）提供企业级数据服务。

#### NetApp SANtricity

NetApp SANtricity 旨在为 E 系列混合闪存和 EF 系列全闪存阵列提供行业领先的性能，可靠性和精简性。为繁重工作负载应用程序（包括数据分析，视频监控以及备份和恢复）实现 E 系列混合闪存和 EF 系列全闪存阵列的最高性能和利用率。借助 SANtricity，可以在存储保持联机状态的同时完成配置调整，维护，容量扩展和其他任务。SANtricity 还提供卓越的数据保护，主动监控和认证安全性—所有这些功能均可通过易于使用的机载 System Manager 界面进行访问。要了解更多信息，请参见 ["NetApp E 系列 SANtricity 软件产品规格"](#)。

#### 性能优化

经过性能优化的 SANtricity 软件可为您的所有数据分析，视频监控和备份应用程序提供具有高 IOPS，高吞吐量和低延迟的数据。提高高 IOPS，低延迟应用程序和高带宽，高吞吐量应用程序的性能。

最大限度地延长正常运行时间

在存储保持联机的情况下完成所有管理任务。在不中断 I/O 的情况下调整配置，执行维护或扩展容量借助自动化功能，联机配置，最先进的动态磁盘池（Dynamic Disk Pool，DPP）技术等实现同类最佳的可靠性。

请轻松休息

SANtricity 软件可通过易于使用的机载 System Manager 界面提供卓越的数据保护，主动监控和认证安全性。简化存储管理任务。获得对所有 E 系列存储系统进行高级调整所需的灵活性。随时随地管理您的 NetApp E 系列系统。我们基于 Web 的盒装界面简化了您的管理工作流。

## NetApp Trident

"Trident" NetApp 是适用于 Docker 和 Kubernetes 的开源动态存储编排程序，可简化永久性存储的创建，管理和使用。Trident 是 Kubernetes 原生应用程序，直接在 Kubernetes 集群中运行。借助 Trident，客户可以将 DL 容器映像无缝部署到 NetApp 存储上，并为 AI 容器部署提供企业级体验。Kubernetes 用户（例如 ML 开发人员和数据科学家）可以创建，管理和自动化流程编排和克隆，从而充分利用 NetApp 技术提供的 NetApp 高级数据管理功能。

## NetApp BlueXP复制和同步

"BlueXP复制和同步" 是一项 NetApp 服务，用于快速安全地同步数据。无论您是需要内部NFS还是SMB文件共享、NetApp StorageGRID、NetApp ONTAP S3、NetApp Cloud Volumes Service、Azure NetApp Files、Amazon Simple Storage Service (Amazon S3)、Amazon Elastic File System (Amazon EFS)、Azure Blob、Google Cloud Storage、或者IBM Cloud Object Storage、BlueXP Copy and Sync可将文件快速安全地移动到您需要的位置。数据传输完成后，即可在源和目标上完全使用。BlueXP复制和同步功能会根据预定义的计划持续同步数据、仅移动增量、从而最大限度地减少数据复制所需的时间和资金。BlueXP Copy and Sync是一款软件即服务(SaaS)工具、设置和使用极其简单。由BlueXP复制和同步触发的数据传输由数据代理执行。您可以在AWS、Azure、Google Cloud Platform或内部部署BlueXP Copy and Sync数据代理。

## 联想 ThinkSystem 服务器

联想 ThinkSystem 服务器采用创新的硬件，软件和服务，可解决客户当前面临的挑战，并提供一种循序渐进的模块化设计方法来应对未来的挑战。这些服务器利用同类最佳的行业标准技术以及联想的差异化创新技术，为 x86 服务器提供最大的灵活性。

部署联想 ThinkSystem 服务器的主要优势包括：

- 高度可扩展的模块化设计，可随业务发展而扩展
- 行业领先的弹性，可节省数小时的成本高昂的计划外停机时间
- 快速闪存技术可实现更低的延迟，更快的响应速度以及更智能的实时数据管理

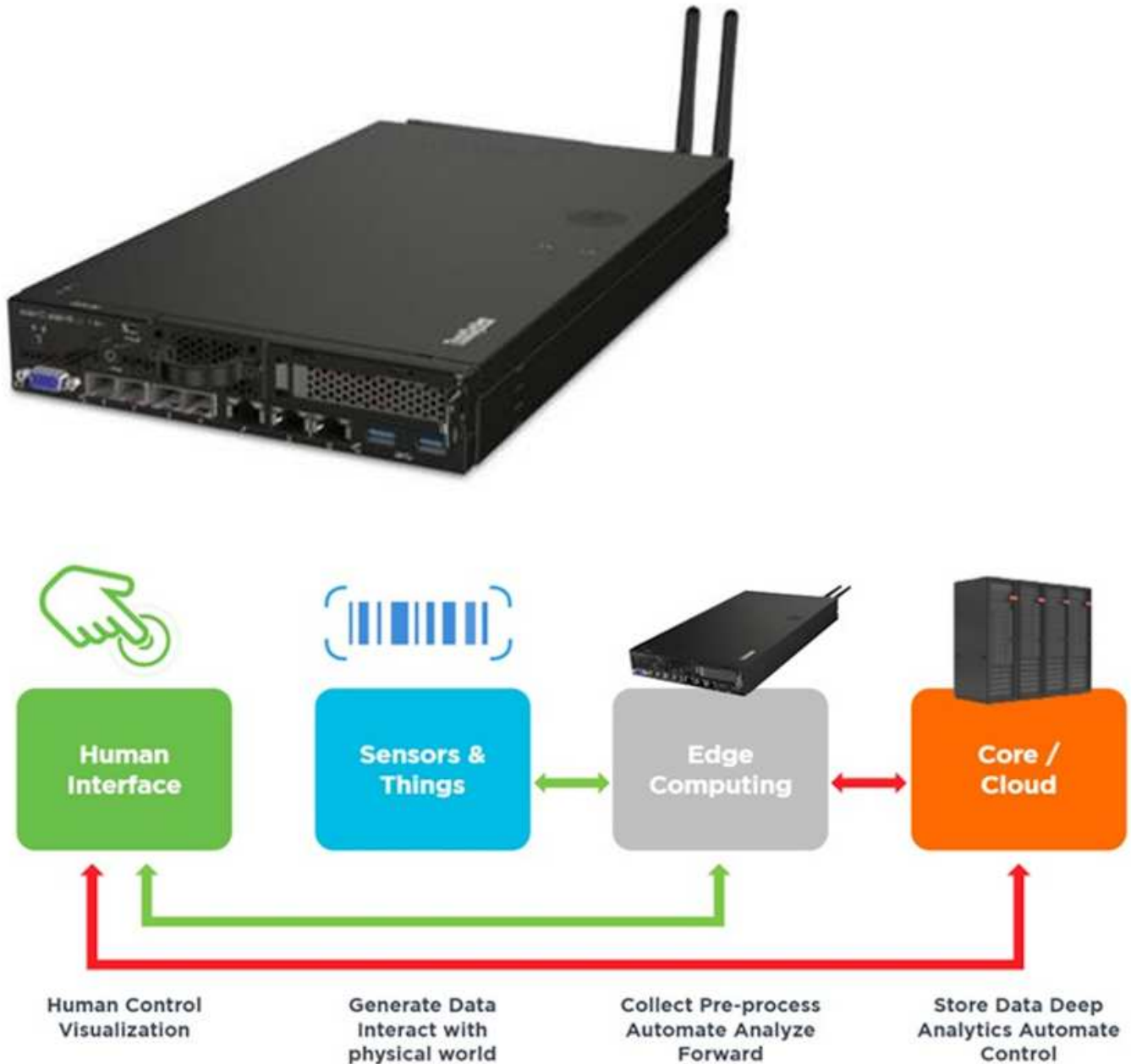
在 AI 领域，联想正在采取切实可行的方法帮助企业了解 ML 和 AI 的优势并将其用于工作负载。联想客户可以在联想 AI 创新中心探索和评估联想 AI 产品，以充分了解其特定用例的价值。为了缩短实现价值的时间，这种以客户为中心的方法可以为客户提供解决方案开发平台的概念验证，这些平台已准备就绪，可供 AI 使用并进行优化。

## 联想 ThinkSystem SE350 边缘服务器

通过边缘计算，可以在将 IoT 设备中的数据发送到数据中心或云之前在网络边缘对其进行分析。如下图所示，联想 ThinkSystem SE350 专为满足边缘部署的独特要求而设计，采用紧凑的加固型环境加固外形，重点关注灵活性，连接性，安全性和远程易管理性。



SE350 采用 Intel Xeon D 处理器，可以灵活地支持边缘 AI 工作负载的加速，专为应对数据中心以外各种环境中服务器部署的挑战而构建。



## MLPerf

MLPerf 是用于评估 AI 性能的行业领先基准套件。它涵盖应用 AI 的许多方面，包括图像分类，对象检测，医学影像和自然语言处理（NLP）。在此验证中，我们使用了推理 v0.7 工作负载，这是此验证完成时 MLPerf 推理的最新迭代。。"MLPerf 推理 v0.7" 套件包括四个适用于数据中心和边缘系统的新基准：

- Transformers（Bert）提供的 \* 双向编码器表示法经过微调，可使用 squad 数据集回答问题。
- \* 深度学习建议模式（DLRM）是一种个性化和建议模式，经过培训可优化点击率（CTR）。
- \* 3D U-Net.\* 3D U-Net 架构接受过有关脑肿瘤分段（Bras）数据集的培训。
- \* RNN-T.\* 经常性神经网络传感器（RNNP-T）是一种自动语音识别（Automatic Speech Recognition，ASR）模型，该模型经过部分 LibriSpeech 的训练。MLPerf 推理结果和代码已公开发布，并已获得 Apache

许可证。MLPerf 推理具有一个 Edge 分支，可支持以下情形：

- \* 单一流。\* 此场景模拟响应能力是关键因素的系统，例如在智能手机上执行脱机 AI 查询。各个查询将发送到系统并记录响应时间。结果将报告所有响应的 90 百分位延迟。
- \* 多流。\* 此基准测试适用于处理来自多个传感器的输入的系统。在测试期间，系统会按固定的时间间隔发送查询。会施加 QoS 限制（允许的最大延迟）。此测试将报告系统在满足 QoS 限制的情况下可以处理的流数量。
- \* 脱机。\* 这是涉及批处理应用程序的最简单情形，指标是每秒样本吞吐量。系统可以使用所有数据，基准测试可测量处理所有样本所需的时间。

联想已发布了本文档中使用的服务器 SE350 与 T4 的 MLPerf 推理得分。有关结果，请参见 ["https://mlperf.org/inference-results-0-7/"](https://mlperf.org/inference-results-0-7/) 在条目 #0.7-145 的 "边缘，封闭分区" 一节中。

测试计划

本文档遵循 MLPerf 推理 v0.7 "code"，MLPerf 推理 v1.1 "code"，和 "rules"。我们运行了专为在边缘进行推理而设计的 MLPerf 基准测试，如下表所述。

区域	任务	型号	数据集	QSL 大小	质量	多流延迟限制
愿景	映像分类	Resnet50v1.5	ImageNet ( 224x224 )	1024	FP32 的 99%	50 毫秒
愿景	对象检测 (大型)	SSD - ResNet34	可可 ( 1200 x 1200 )	64	FP32 的 99%	66 毫秒
愿景	对象检测 (小型)	SSD — MobileNetsv1	可可 ( 300 x 300 )	256	FP32 的 99%	50 毫秒
愿景	医学影像分段	3D UNET	Brats 2019 ( 224x224x160 )	16.	FP32 的 99% 和 99.9%	不适用
语音	语音到文本	RNNT	Libraispeech 开发清理	2513	FP32 的 99%	不适用
language	语言处理	Bert	Sad v1.1	10833	FP32 的 99%	不适用

下表介绍了 Edge 基准测试场景。

区域	任务	场景
愿景	映像分类	单流，脱机，多流
愿景	对象检测 (大型)	单流，脱机，多流
愿景	对象检测 (小型)	单流，脱机，多流
愿景	医学影像分段	单个流，脱机
语音	语音到文本	单个流，脱机
language	语言处理	单个流，脱机

我们使用在此验证中开发的网络存储架构执行了这些基准测试，并将结果与先前提交给 MLPerf 的边缘服务器上本地运行的结果进行了比较。比较结果是，确定共享存储对推理性能有多大影响。

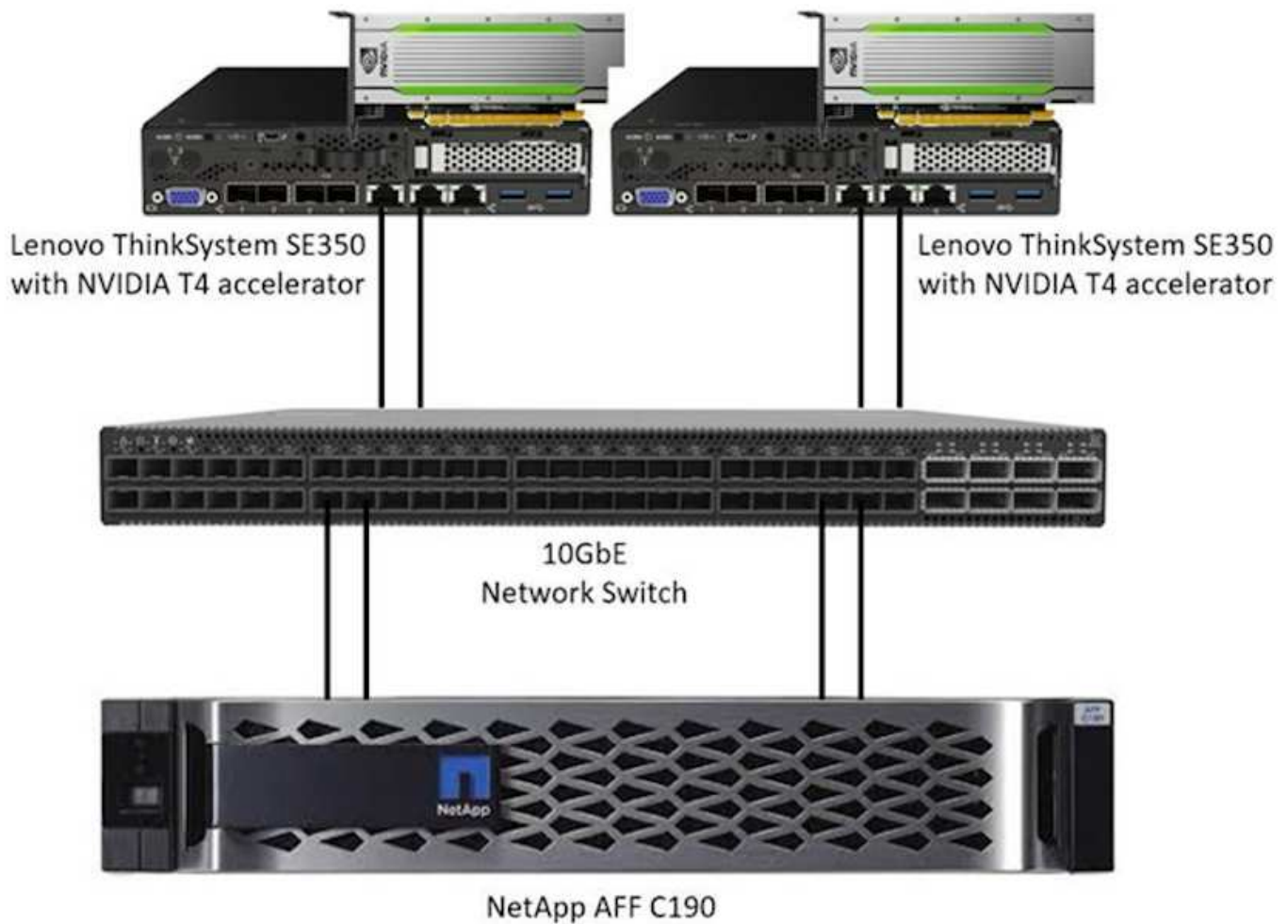
## 测试配置

下图显示了测试配置。我们使用了 NetApp AFF C190 存储系统和两个联想 ThinkSystem SE350 服务器（每个服务器都有一个 NVIDIA T4 加速器）。这些组件通过 10GbE 网络交换机进行连接。网络存储包含验证 / 测试数据集和经过预先培训的模型。这些服务器可提供计算功能，并可通过 NFS 协议访问存储。

本节介绍测试的配置，网络基础架构， SE350 服务器以及存储配置详细信息。下表列出了解决方案架构的基本组件。

解决方案组件	详细信息
联想 ThinkSystem 服务器	<ul style="list-style-type: none"><li>• 2 个 SE350 服务器，每个服务器具有一个 NVIDIA T4 GPU 卡</li></ul>
	<ul style="list-style-type: none"><li>• 每个服务器都包含一个 Intel Xeon D-2123IT CPU ，其中四个物理核心运行速度为 2.20GHz ，并具有 128 GB RAM</li></ul>
入门级 NetApp AFF 存储系统（ HA 对）	<ul style="list-style-type: none"><li>• NetApp ONTAP 9 软件</li><li>• 24 个 960 GB SSD</li><li>• NFS 协议</li><li>• 每个控制器一个接口组，挂载点有四个逻辑 IP 地址</li></ul>





下表列出了存储配置：具有 2RU 的 AFF C190 ， 24 个驱动器插槽。

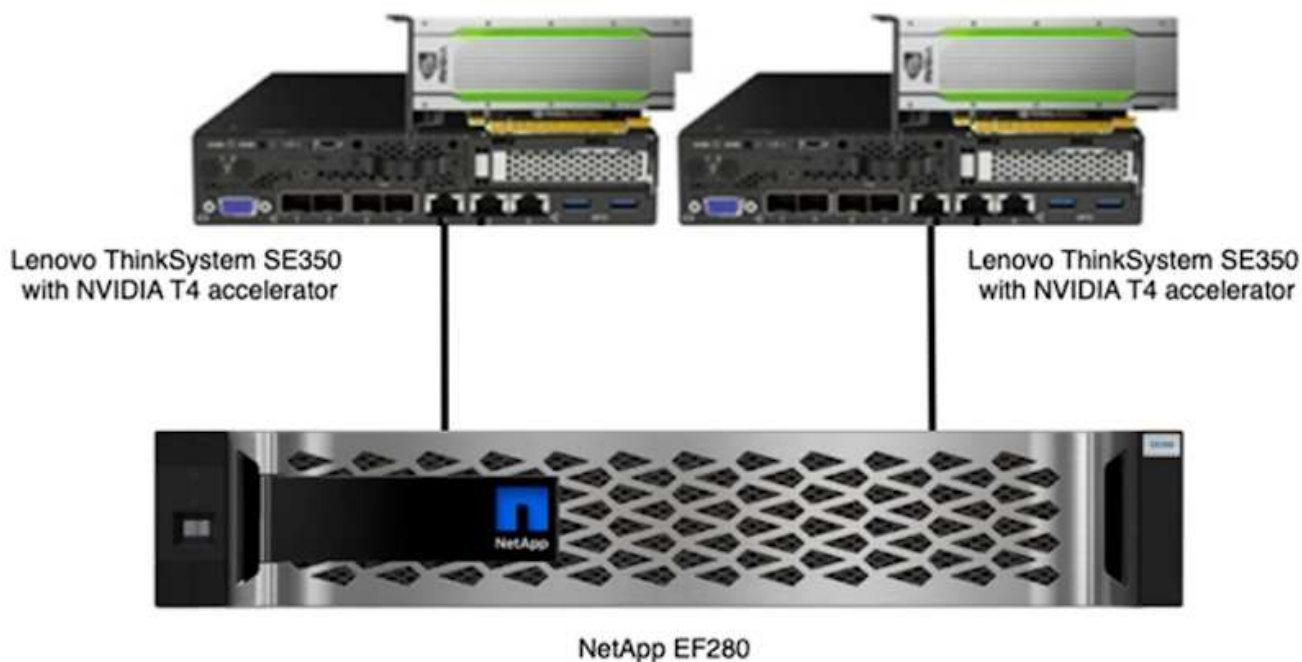
控制器	聚合	FlexGroup 卷	聚合大小	卷大小	操作系统挂载点
控制器 1	aggr1.	/netapplenovo_AI_fg	8.42 TiB	15 TB	/NetApp_Lenovo, fg
控制器 2.	Aggr2.		8.42 TiB		

/netappLenovo AI\_fg 文件夹包含用于模型验证的数据集。

下图显示了测试配置。我们使用了 NetApp EF280 存储系统和两个联想 ThinkSystem SE350 服务器（每个服务器都有一个 NVIDIA T4 加速器）。这些组件通过 10GbE 网络交换机进行连接。网络存储包含验证 / 测试数据集和经过预先培训的模型。这些服务器可提供计算功能，并可通过 NFS 协议访问存储。

下表列出了 EF280 的存储配置。

控制器	卷组	Volume	卷大小	DDP 大小	连接方法
控制器 1	DDP1	卷 1	8.42 TiB	16 TB	SE350-1 到 iSCSI LUN 0
控制器 2.		第 2 卷	8.42 TiB		SE350-2 到 iSCSI LUN 1



## 测试操作步骤

本节介绍用于验证此解决方案的测试过程。

### 操作系统和 AI 推理设置

对于 AFF C190，我们使用支持 NVIDIA GPU 的 Ubuntu 18.04 和 Docker，并使用 MLPerf ["code"](#) 在联想提交到 MLPerf 推理 v0.7 的过程中提供。

对于 EF280，我们将 Ubuntu 20.04 与 NVIDIA 驱动程序结合使用，并将 Docker 与 NVIDIA GPU 和 MLPerf 结合使用 ["code"](#) 在联想提交到 MLPerf 推理 v1.1 的过程中提供。

要设置 AI 推理，请执行以下步骤：

1. 下载需要注册的数据集，ImageNet 2012 验证集，Criteo TB 数据集和 Brats 2019 训练集，然后解压缩这些文件。
2. 创建一个至少包含 1 TB 的工作目录，并定义环境变量 `MLPERF_scrating_path` 引用该目录。

在使用本地数据进行测试时，您应在共享存储上共享此目录以供网络存储使用情形使用，或者在本地磁盘上共享此目录。

3. 运行 `make prebuild` 命令，该命令可为所需的推理任务构建并启动 Docker 容器。



以下命令全部从正在运行的 Docker 容器中执行：

- 为 MLPerf 推理任务下载经过预先培训的 AI 型号：`make download_model`
- 下载可免费下载的其他数据集：`make download_data`

- 预处理数据： `make preprocess_data`
- 运行： `m` 构建。
- 针对计算服务器中的 GPU 优化的构建推理引擎： `make generate_engine`
- 要运行推理工作负载，请运行以下命令（一个命令）：

```
make run_harness RUN_ARGS="--benchmarks=<BENCHMARKS>
--scenarios=<SCENARIOS>"
```

## AI 推理运行

执行了三种类型的运行：

- 使用本地存储的单服务器 AI 推理
- 使用网络存储的单服务器 AI 推理
- 使用网络存储的多服务器 AI 推理

## 测试结果

我们会运行大量测试来评估建议的架构的性能。

有六种不同的工作负载（图像分类，对象检测（小），对象检测（大），医学影像，语音到文本，和自然语言处理（NLP）），您可以在三种不同的情形下运行：脱机，单流和多流。



最后一种情形仅适用于映像分类和对象检测。

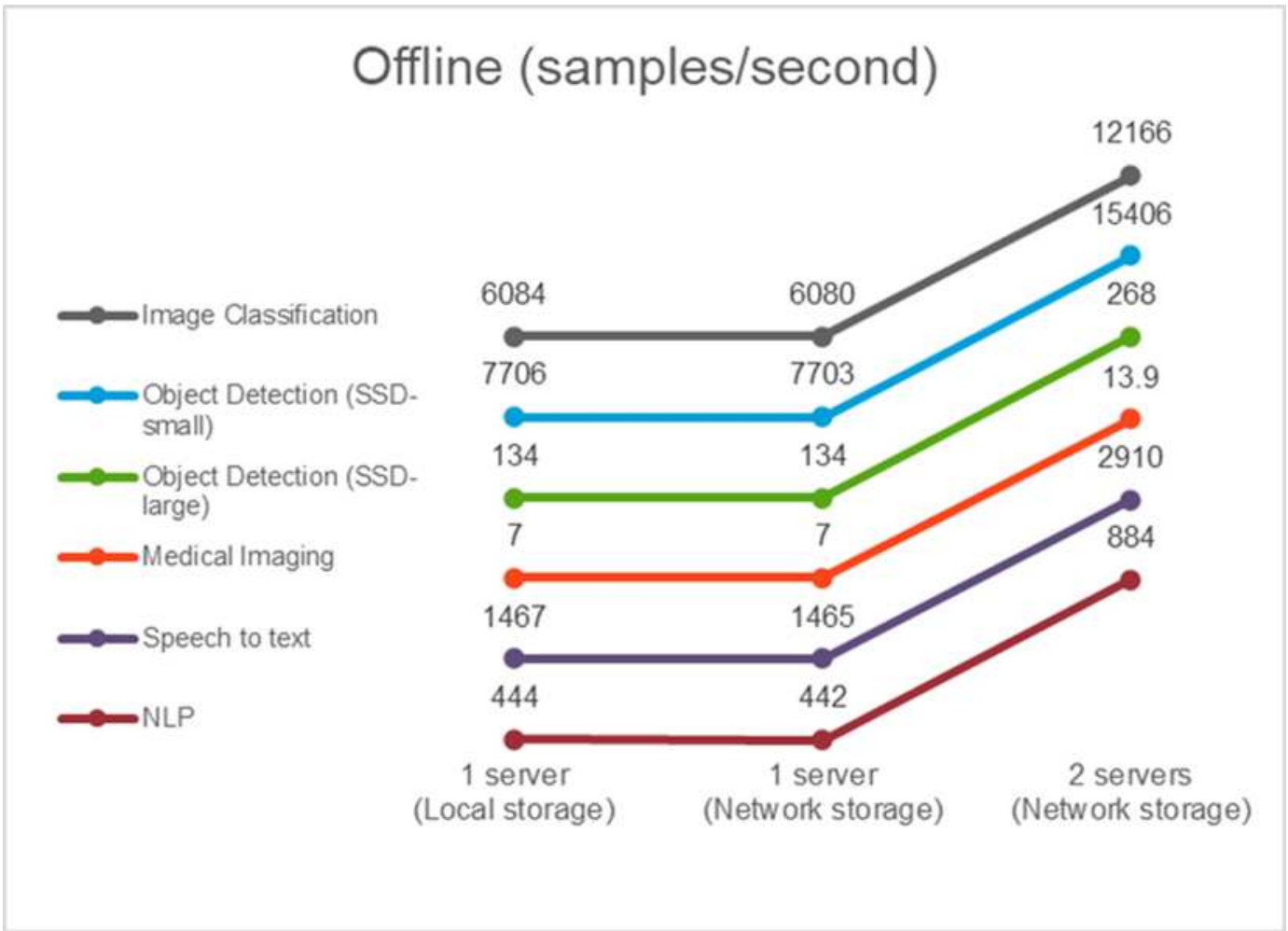
这样可以提供 15 个可能的工作负载，这些工作负载都在三种不同的设置下进行了测试：

- 单个服务器 / 本地存储
- 单个服务器 / 网络存储
- 多服务器 / 网络存储

以下各节将介绍这些结果。

### AFF 脱机情形中的 AI 推理

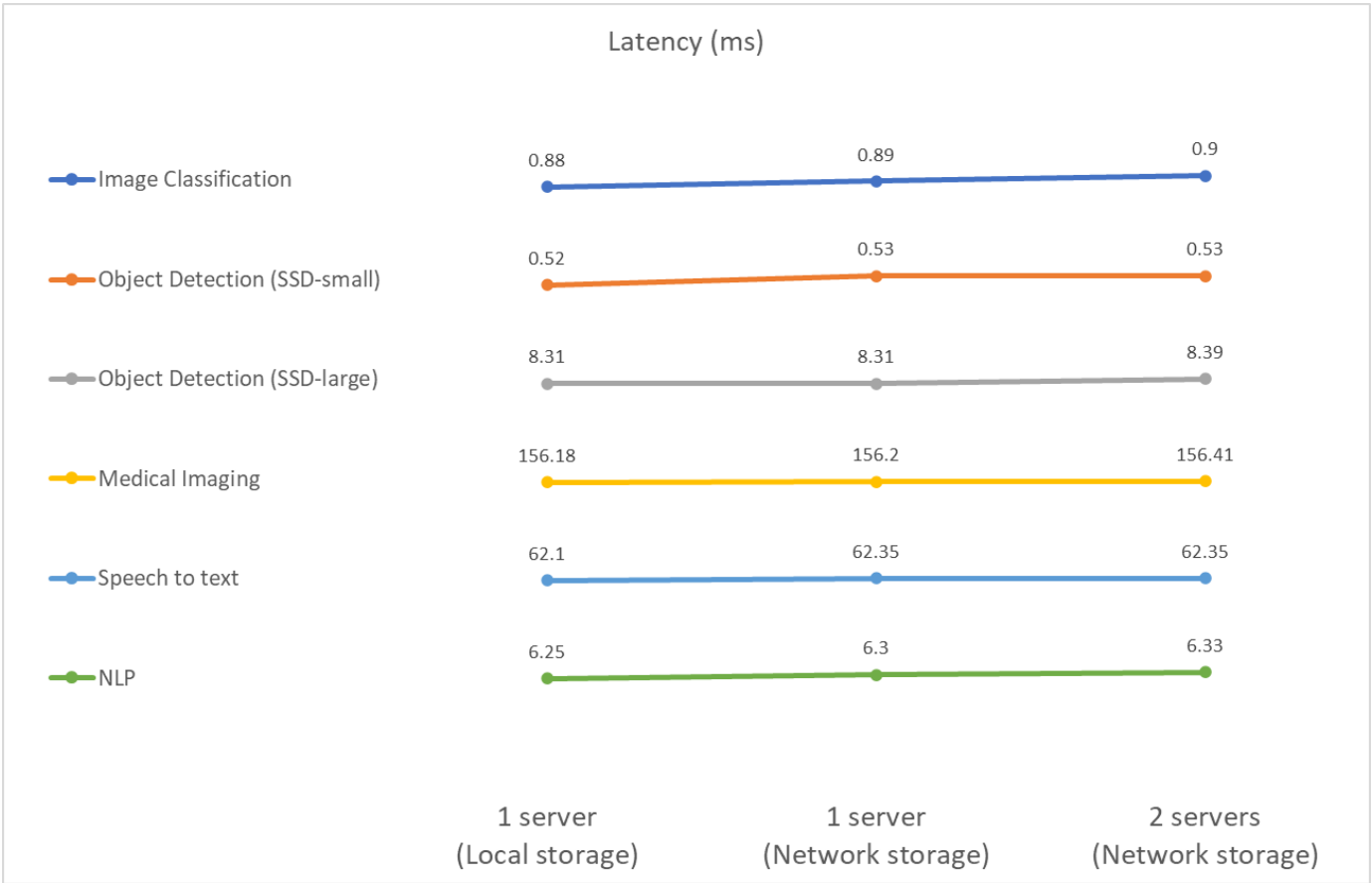
在这种情况下，服务器可以使用所有数据，并测量了处理所有样本所需的时间。我们会将带宽报告为每秒样本数作为测试结果。如果使用了多个计算服务器，则会报告所有服务器的总带宽总和。下图显示了所有这三种使用情形的结果。对于双服务器情形，我们会报告两个服务器的总带宽。



结果显示，网络存储不会对性能产生负面影响，更改极少，对于某些任务，未找到任何结果。添加第二台服务器时，总带宽恰好是两倍，或者最差情况下，更改率小于 1%。

#### 在 **AFF** 的单个流方案中进行 **AI** 推理

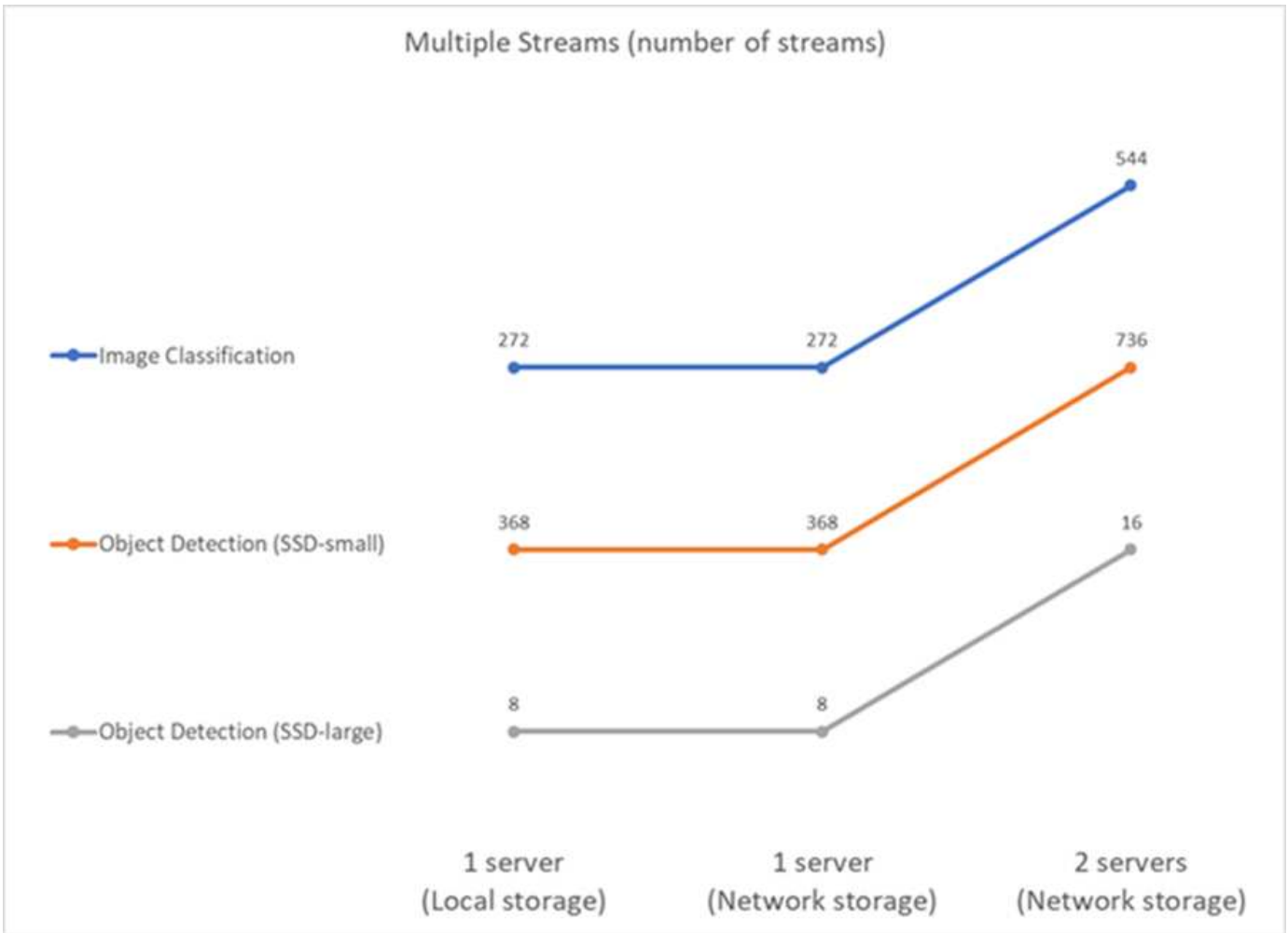
此基准测试可测量延迟。对于多个计算服务器案例，我们会报告平均延迟。下图显示了这组任务的结果。对于双服务器案例，我们会报告两个服务器的平均延迟。



结果再次表明，网络存储足以处理这些任务。在一台服务器的情况下，本地存储与网络存储之间的差别很小或没有差别。同样，当两个服务器使用相同的存储时，两个服务器上的延迟保持不变或变化量非常小。

在 **AFF** 的多流方案中进行 **AI** 推理

在这种情况下，结果是系统在满足 QoS 限制的情况下可以处理的流数量。因此，结果始终为整数。对于多个服务器，我们会报告所有服务器上的流总数。并非所有工作负载都支持此方案，但我们已执行了这些工作负载。下图总结了我们的测试结果。对于双服务器案例，我们会报告两个服务器的流总数。



结果显示了设置的完美性能—本地存储和网络存储的结果相同，添加第二个服务器会使建议设置可以处理的流数量增加一倍。

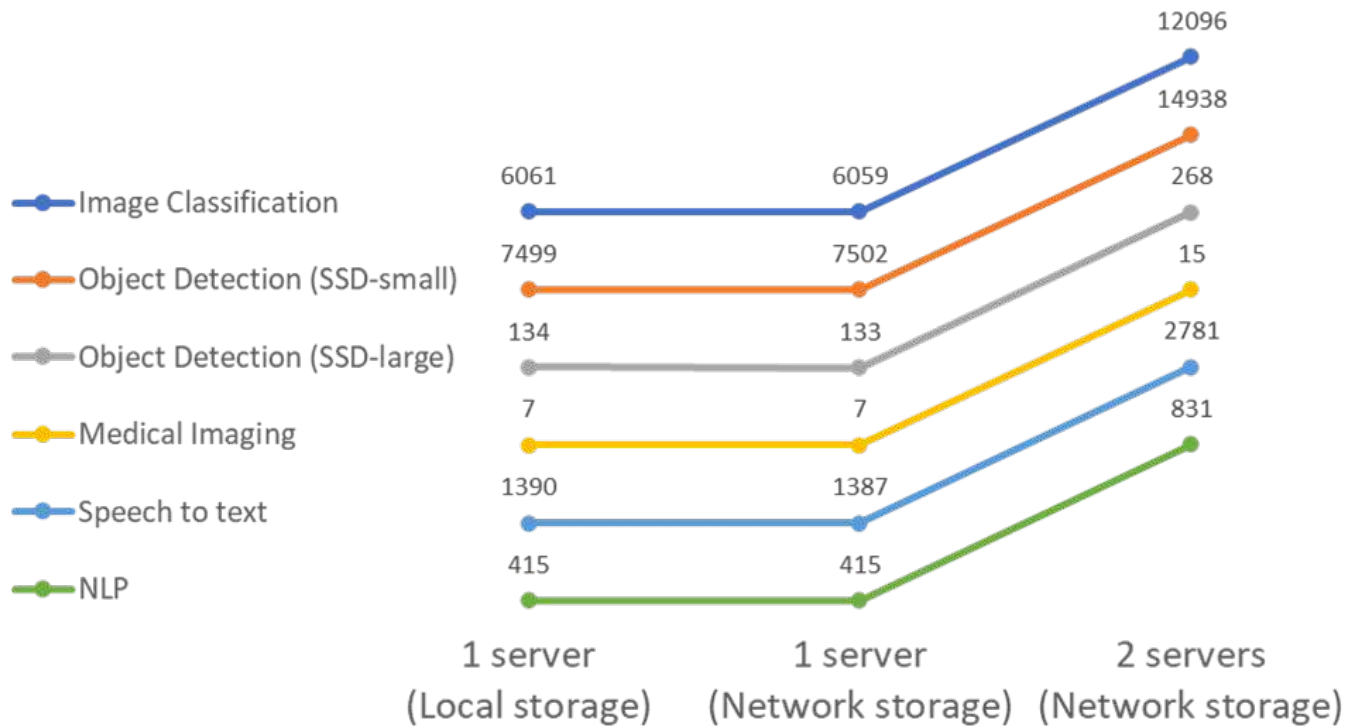
### EF 的测试结果

我们会运行大量测试来评估建议的架构的性能。有六种不同的工作负载（图像分类，对象检测（小），对象检测（大），医学影像，语音到文本，和自然语言处理（NLP）），这两种情况下运行：脱机和单流。以下各节将介绍这些结果。

#### EF 脱机情形中的 AI 推理

在这种情况下，服务器可以使用所有数据，并测量了处理所有样本所需的时间。我们会将带宽报告为每秒样本数作为测试结果。对于单节点运行，我们会报告两个服务器的平均值，而对于两个服务器运行，我们会报告所有服务器的总带宽总和。下图显示了使用情形的结果。

## Offline (samples/second)

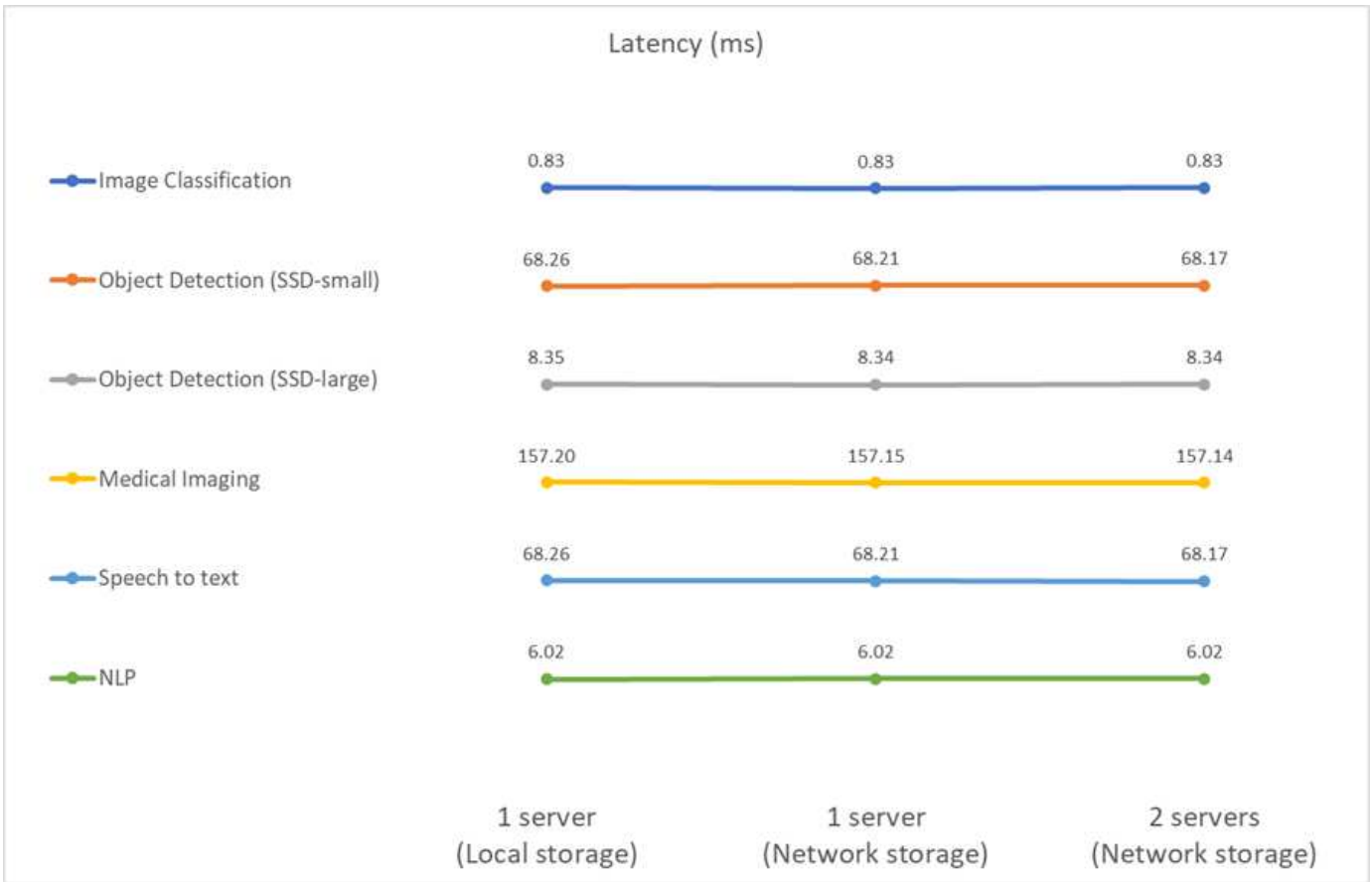


结果显示，网络存储不会对性能产生负面影响，更改极少，对于某些任务，未找到任何结果。添加第二台服务器时，总带宽恰好是两倍，或者最差情况下，更改率小于 1%。

在一个流场景中对 **EF** 进行 **AI** 推理

此基准测试可测量延迟。对于所有情况，我们都会报告运行中涉及的所有服务器的平均延迟。系统将提供此任务套件的结果。





结果再次显示，网络存储足以处理这些任务。在一台服务器的情况下，本地存储与网络存储之间的差别很小或没有差别。同样，当两个服务器使用相同的存储时，两个服务器上的延迟保持不变或变化量非常小。

## 架构规模估算选项

您可以调整用于验证的设置，以适合其他使用情形。

### 计算服务器

我们使用的是 Intel Xeon D-2123IT CPU，这是 SE350 支持的最低 CPU 级别，具有四个物理核心和 60 瓦 TDP。虽然服务器不支持更换 CPU，但可以订购功能更强大的 CPU。支持的最大 CPU 是 Intel Xeon D-2183IT，16 核，100 瓦，运行频率为 2.20 GHz。这样可以显著提高 CPU 计算能力。虽然 CPU 本身并不是运行推理工作负载的瓶颈，但它有助于处理数据以及执行与推理相关的其他任务。目前，NVIDIA T4 是唯一可用于边缘使用情形的 GPU；因此，目前无法升级或降级 GPU。

### 共享存储

在测试和验证中，我们在本文档中使用了 NetApp AFF C190 系统，该系统的最大存储容量为 50.5 TB，连续读取的吞吐量为 4.4 GBps，小型随机读取的 IOPS 为 230k，经证实非常适合边缘推理工作负载。

但是、如果您需要更多存储容量或更快的网络速度、则应使用 NetApp AFF A220 或 NetApp AFF A250 存储系统。此外，NetApp EF280 系统的最大容量为 1.5 PB，带宽为 10 Gbps，也用于此解决方案验证。如果您希望使用更多的存储容量和更高的带宽、["NetApp EF300"](#) 可以使用。

## 结论

AI 驱动的自动化和边缘计算是帮助业务组织实现数字化转型并最大限度地提高运营效率和安全性的领先方法。借助边缘计算，数据处理速度更快，因为它不必往返于数据中心。因此，将数据来回发送到数据中心或云的相关成本会降低。如果企业必须使用部署在边缘的人工智能推理模型近乎实时地做出决策，则降低延迟和提高速度将非常有用。

NetApp 存储系统可提供与本地 SSD 存储相同或更好的性能，并为数据科学家，数据工程师， AI/ML 开发人员以及业务或 IT 决策者带来以下优势：

- 在 AI 系统，分析和其他关键业务系统之间轻松共享数据。这种数据共享可减少基础架构开销，提高性能并简化整个企业的数据管理。
- 可独立扩展的计算和存储，最大限度地降低成本并提高资源利用率。
- 利用集成的 Snapshot 副本和克隆简化开发和部署工作流，实现瞬时且节省空间的用户工作空间，集成版本控制和自动化部署。
- 企业级数据保护，可实现灾难恢复和业务连续性。本文档中介绍的 NetApp 和联想解决方案是一种灵活的横向扩展架构，非常适合在边缘进行企业级 AI 推理部署。

## 致谢

- J.Falkanger ，高级联想 HPC & AI 解决方案经理
- NetApp 技术营销工程师 Dave Arnette
- NetApp E 系列 AI 解决方案技术主管 Joey Parnell
- NetApp QA 工程师 Cody Harryman

## 从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请参见以下文档和 / 或网站：

- NetApp AFF A 系列阵列产品页面  
["https://www.netapp.com/data-storage/aff-a-series/"](https://www.netapp.com/data-storage/aff-a-series/)
- NetApp ONTAP 数据管理软件— ONTAP 9 信息库  
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>
- TR-4727 ： 《 NetApp EF 系列简介》  
<https://www.netapp.com/pdf.html?item=/media/17179-tr4727pdf.pdf>
- NetApp E 系列 SANtricity 软件产品规格  
<https://www.netapp.com/pdf.html?item=/media/19775-ds-3171-66862.pdf>
- 适用于容器的 NetApp 持久存储— NetApp Trident  
["https://netapp.io/persistent-storage-provisioner-for-kubernetes/"](https://netapp.io/persistent-storage-provisioner-for-kubernetes/)

- MLPerf
  - ["https://mlcommons.org/en/"](https://mlcommons.org/en/)
  - ["http://www.image-net.org/"](http://www.image-net.org/)
  - ["https://mlcommons.org/en/news/mlperf-inference-v11/"](https://mlcommons.org/en/news/mlperf-inference-v11/)
- NetApp BlueXP复制和同步  
["https://docs.netapp.com/us-en/occm/concept\\_cloud\\_sync.html#how-cloud-sync-works"](https://docs.netapp.com/us-en/occm/concept_cloud_sync.html#how-cloud-sync-works)
- TensorFlow 基准测试  
["https://github.com/tensorflow/benchmarks"](https://github.com/tensorflow/benchmarks)
- 联想 ThinkSystem SE350 边缘服务器  
["https://lenovopress.com/lp1168"](https://lenovopress.com/lp1168)
- 联想 ThinkSystem DM5100F 统一闪存存储阵列  
["https://lenovopress.com/lp1365-thinksystem-dm5100f-unified-flash-storage-array"](https://lenovopress.com/lp1365-thinksystem-dm5100f-unified-flash-storage-array)

## WP-7328：使用 NVIDIA JarVis 的 NetApp 对话 AI

Rick Huang， Sung-Han Lin， NetApp， NVIDIA 公司的 Dide Onofino

NVIDIA DGX 系统系列由全球首款基于人工智能（AI）的集成系统组成，这些系统专为企业 AI 而构建。NetApp AFF 存储系统可提供极致性能和行业领先的混合云数据管理功能。NetApp 和 NVIDIA 合作创建了 NetApp ONTAP AI 参考架构，这是一款适用于人工智能和机器学习（ML）的统包解决方案工作负载，可提供企业级性能，可靠性和支持。

本白皮书为客户构建对话式 AI 系统提供了方向性指导，以支持各个行业的不同使用情形。其中包括有关使用 NVIDIA JarVis 部署系统的信息。这些测试是使用 NVIDIA DGX 工作站和 NetApp AFF A220 存储系统执行的。

解决方案的目标受众包括以下组：

- 企业架构师，负责设计用于开发人工智能模型和软件的解决方案，用于虚拟零售助理等人工智能对话用例
- 寻求高效方式实现语言建模开发目标的数据科学家
- 负责维护和处理客户问题和对话记录等文本数据的数据工程师
- 有兴趣转变对话式 AI 体验并加快 AI 计划上市速度的高管和 IT 决策者以及业务主管

### 解决方案概述

本文档介绍了适用于 ONTAP AI 和 NVIDIA DGX 的对话式 AI 模型。

### NetApp ONTAP AI 和 BlueXP 复制和同步

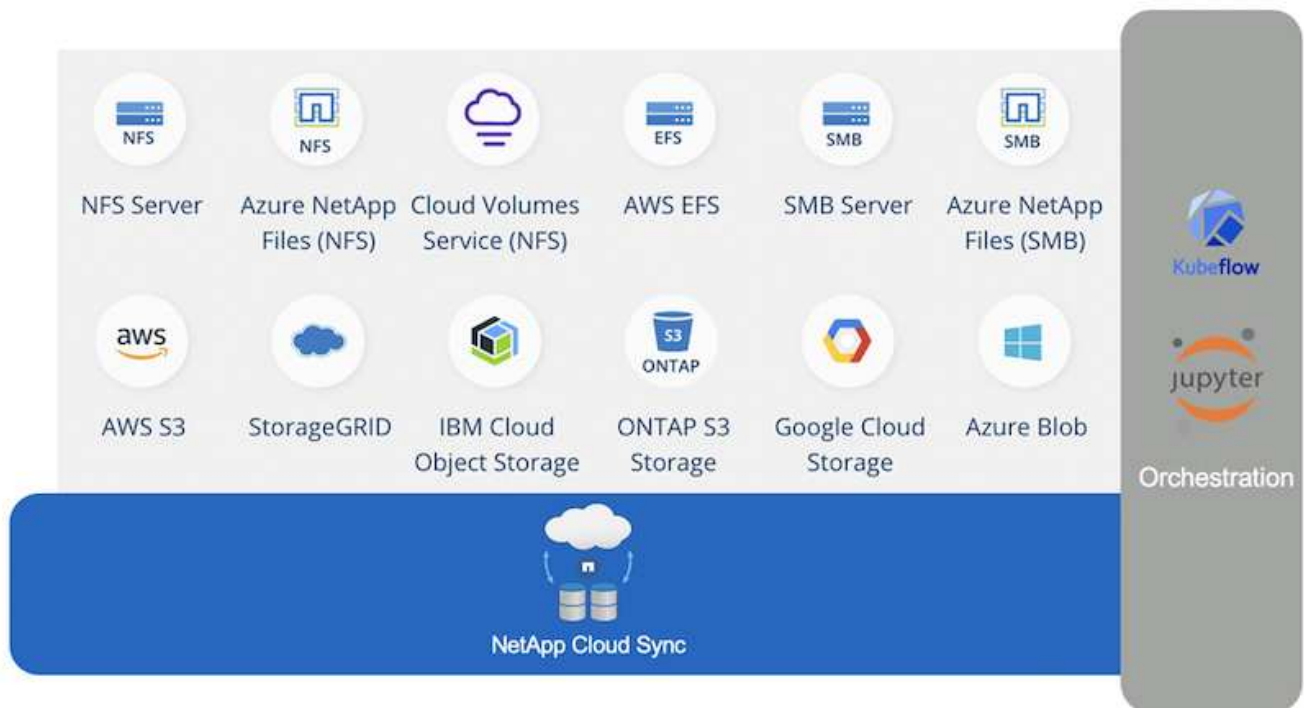
由 NVIDIA DGX 系统和 NetApp 云连接存储系统提供支持的 NetApp ONTAP AI 架构由 NetApp 和 NVIDIA 开发并验证。此参考架构为 IT 组织提供了以下优势：

- 消除设计复杂性
- 支持独立扩展计算和存储
- 支持客户从小规模入手，无缝扩展
- 为各种性能和成本点提供一系列存储选项 NetApp ONTAP AI 可将 DGX 系统和 NetApp AFF A220 存储系统与一流的网络紧密集成在一起。NetApp ONTAP AI 和 DGX 系统消除了设计复杂性和猜测性工作，从而简化了 AI 部署。客户可以从小规模入手，无中断地扩展系统，同时智能地管理从边缘到核心再到云再到云的数据。

借助NetApp BlueXP复制和同步功能、您可以通过各种协议轻松移动数据、无论是在两个NFS共享、两个CIFS共享之间、还是在一个文件共享与Amazon S3、Amazon Elastic File System (EFS)或Azure Blob存储之间。主动 - 主动操作意味着您可以继续同时使用源和目标，并在需要时逐步同步数据更改。BlueXP复制和同步功能支持您在任何源系统和目标系统(无论是内部系统还是基于云的系统)之间移动和增量同步数据、为您使用数据开辟了多种新方式。在内部系统，云入网和云迁移或协作和数据分析之间迁移数据变得非常容易。下图显示了可用的源和目标。

在对话式AI系统中、开发人员可以利用BlueXP Copy and Sync将对话历史记录从云归档到数据中心、以便对自然语言处理(NLP)模型进行离线训练。通过培训模式识别更多意向，对话式 AI 系统将更好地处理最终用户提出的更复杂的问题。

### NVIDIA Jarvis 多模式框架



"NVIDIA Jarvis" 是一个端到端框架，用于构建对话式 AI 服务。它包括以下经过 GPU 优化的服务：

- 自动语音识别（ Automatic Speech Recognition ， As1 ）
- 自然语言理解（ NLF ）
- 与域特定的履行服务集成
- 文本语音转换（ TTS- 语音转换）
- 基于计算机视觉（ CV ） Jarvis 的服务使用最先进的深度学习模型来应对实时对话 AI 这一复杂且极具挑战

性的任务。要与最终用户进行实时自然的交互，模型需要在 300 毫秒内完成计算。自然交互具有挑战性，需要多模式感知集成。模型管道也很复杂，需要在上述服务之间进行协调。

JarVis 是一个完全加速的应用程序框架，用于构建使用端到端深度学习管道的多模式对话 AI 服务。JARVIS 框架包括经过预先培训的人工智能对话模型，工具以及针对语音，视觉和 NLU 任务优化的端到端服务。除了 AI 服务之外，JarVis 还支持您同时融合视觉，音频和其他传感器输入，以便在虚拟助手，多用户化和呼叫中心助理等应用程序中提供多用户，多上下文对话等功能。

## NVIDIA Nemo

"NVIDIA Nemo" 是一款开源 Python 工具包，用于使用易于使用的应用程序编程接口（API）构建，培训和微调 GPU 加速的一流对话 AI 模型。Nemo 使用 NVIDIA GPU 中的 Tensor 核心运行混合精度计算，并可轻松扩展到多个 GPU，以提供尽可能高的训练性能。Nemo 用于为视频呼叫记录，智能视频助理以及医疗保健，金融，零售和电信等不同行业的自动呼叫中心支持等实时应用程序构建模型。

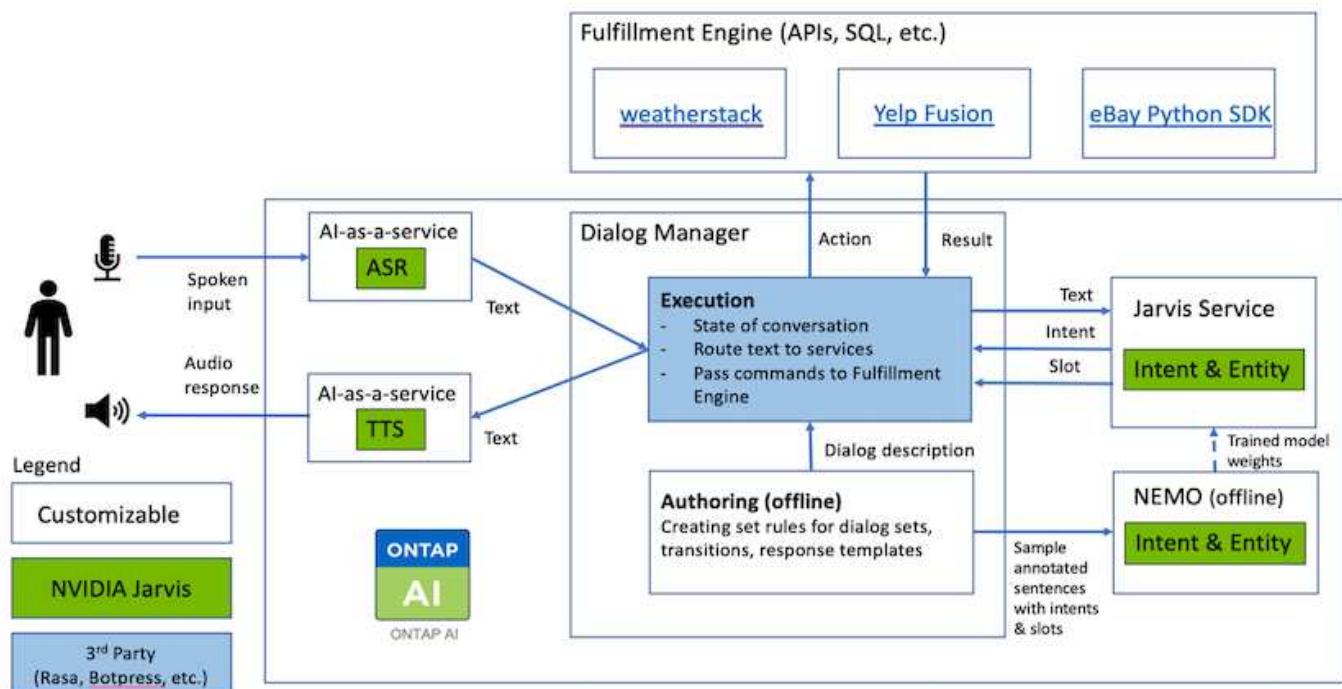
我们使用 Nemo 来训练模型，以便识别归档对话历史记录中用户问题的复杂意图。此培训将零售虚拟助手的功能扩展到了 Jarvis 所提供的功能之外。

### 零售用例摘要

我们使用 NVIDIA Jarvis 构建了一个虚拟零售助理，可接受语音或文本输入并回答有关天气，关注点和库存定价的问题。对话式 AI 系统能够记住对话流，例如，如果用户未指定天气或感兴趣点的位置，可以询问跟进问题。系统还可以识别诸如 "泰国食品" 或 "笔记本电脑内存" 等复杂实体。它了解自然语言问题，例如 "下星期在洛杉矶会下雨吗？" 有关零售虚拟助手的演示，请参见 ["针对零售用例自定义状态和流程"](#)。

## 解决方案技术

下图显示了建议的对话 AI 系统架构。您可以使用语音信号或文本输入与系统进行交互。如果检测到语音输入，则 JARVIS AI as-service（AlaaS）将执行 As1，以便为对话框管理器生成文本。对话框管理器会记住对话状态，将文本路由到相应的服务，并将命令传递到实施引擎。JARVIS NLP 服务会输入文本，识别意向和实体，并将这些意向和实体插槽输出回对话框管理器，然后由该对话框管理器向执行引擎发送操作。履行引擎由问题解答用户查询的第三方 API 或 SQL 数据库组成。从实施引擎收到结果后，对话管理器会将文本路由到 JarVis TTSAaaS，以便为最终用户生成音频响应。我们可以归档对话历史记录，为 Nemo 培训添加意向和插槽，以便随着更多用户与系统交互，NLP 服务得到改进。



### 硬件要求

此解决方案已通过一个 DGX 工作站和一个 AFF A220 存储系统的验证。JARVIS 需要使用 T4 或 V100 GPU 来执行深度神经网络计算。

下表列出了在测试中实施解决方案所需的硬件组件。

硬件	数量
T4 或 V100 GPU	1.
NVIDIA DGX Station	1.

### 软件要求

下表列出了在测试中实施解决方案所需的软件组件。

软件	版本或其他信息
NetApp ONTAP 数据管理软件	9.6
Cisco NX-OS 交换机固件	7.0 (3) I6 (1)
NVIDIA DGX 操作系统	4.0.4 — Ubuntu 18.04 LTS
NVIDIA JarVis Framework	EA v0.2
NVIDIA Nemo	nvc.io/nvidia/nemo : v0.10
Docker 容器平台	18.06.1-ce [e68fc7a]



## 概述

本节详细介绍了虚拟零售助理的实施。

### JarVis 部署

您可以注册 "[JARVIS 早期访问计划](#)" 访问 NVIDIA GPU Cloud (NGC) 上的 JarVis 容器。从 NVIDIA 收到凭据后，您可以使用以下步骤部署 JarVis：

1. 登录到 NGC。
2. 在 NGC 上设置您的组织：ea-2-JarVis。
3. 找到 JarVis EA v0.2 资产：JarVis containers are in Private Registry > Organization Containers。
4. 选择 JarVis：导航到 Model Scripts，然后单击 JarVis Quick Start
5. 验证所有资产是否均正常工作。
6. 查找用于构建您自己的应用程序的文档：PDF 可在 Model Scripts > JarVis Documentation > File Browser 中找到。

### 针对零售用例自定义状态和流程

您可以针对特定使用情形自定义对话框管理器的状态和流。在我们的零售示例中，我们提供了以下四个 YAML 文件，用于根据不同意向指导对话。

查看以下文件名列表以及每个文件的问题描述：

- main\_flow.yml：定义主要对话流和状态，并在必要时将此流定向到其他三个 YAML 文件。
- retail\_flow.yml：包含与零售或感兴趣点问题相关的状态。系统会提供最近商店的信息或给定商品的价格。
- weather\_flow.yml：包含与天气问题相关的状态。如果无法确定位置，系统会询问一个跟进问题以进行澄清。
- error\_flow.yml：处理用户意向不属于上述三个 YAML 文件的情况。显示错误消息后，系统会重新路由到接受用户问题。以下各节包含这些 YAML 文件的详细定义。

#### main\_flow.yml

```
name: JarvisRetail
intent_transitions:
  jarvis_error: error
  price_check: retail_price_check
  inventory_check: retail_inventory_check
  store_location: retail_store_location
  weather.weather: weather
  weather.temperature: temperature
  weather.sunny: sunny
  weather.cloudy: cloudy
```



```

weather.snow: snow
weather.rainfall: rain
weather.snow_yes_no: snowfall
weather.rainfall_yes_no: rainfall
weather.temperature_yes_no: tempyesno
weather.humidity: humidity
weather.humidity_yes_no: humidity
navigation.startnavigationpoi: retail # Transitions should be context
and slot based. Redirecting for now.
navigation.geteta: retail
navigation.showdirection: retail
navigation.showmappoi: idk_what_you_talkin_about
nomatch.none: idk_what_you_talkin_about
states:
  init:
    type: message_text
    properties:
      text: "Hi, welcome to NARA retail and weather service. How can I
help you?"
    input_intent:
      type: input_context
      properties:
        nlp_type: jarvis
        entities:
          intent: dontcare
# This state is executed if the intent was not understood
dont_get_the_intent:
  type: message_text_random
  properties:
    responses:
      - "Sorry I didn't get that! Please come again."
      - "I beg your pardon! Say that again?"
      - "Are we talking about weather? What would you like to know?"
      - "Sorry I know only about the weather"
      - "You can ask me about the weather, the rainfall, the
temperature, I don't know much more"
    delay: 0
    transitions:
      next_state: input_intent
  idk_what_you_talkin_about:
    type: message_text_random
    properties:
      responses:
        - "Sorry I didn't get that! Please come again."
        - "I beg your pardon! Say that again?"
        - "Are we talking about retail or weather? What would you like to

```

```

know?"
    - "Sorry I know only about retail and the weather"
    - "You can ask me about retail information or the weather, the
rainfall, the temperature. I don't know much more."
    delay: 0
    transitions:
        next_state: input_intent
error:
    type: change_context
    properties:
        update_keys:
            intent: 'error'
    transitions:
        flow: error_flow
retail_inventory_check:
    type: change_context
    properties:
        update_keys:
            intent: 'retail_inventory_check'
    transitions:
        flow: retail_flow
retail_price_check:
    type: change_context
    properties:
        update_keys:
            intent: 'check_item_price'
    transitions:
        flow: retail_flow
retail_store_location:
    type: change_context
    properties:
        update_keys:
            intent: 'find_the_store'
    transitions:
        flow: retail_flow
weather:
    type: change_context
    properties:
        update_keys:
            intent: 'weather'
    transitions:
        flow: weather_flow
temperature:
    type: change_context
    properties:
        update_keys:

```

```

        intent: 'temperature'
    transitions:
        flow: weather_flow
rainfall:
    type: change_context
    properties:
        update_keys:
            intent: 'rainfall'
    transitions:
        flow: weather_flow
sunny:
    type: change_context
    properties:
        update_keys:
            intent: 'sunny'
    transitions:
        flow: weather_flow
cloudy:
    type: change_context
    properties:
        update_keys:
            intent: 'cloudy'
    transitions:
        flow: weather_flow
snow:
    type: change_context
    properties:
        update_keys:
            intent: 'snow'
    transitions:
        flow: weather_flow
rain:
    type: change_context
    properties:
        update_keys:
            intent: 'rain'
    transitions:
        flow: weather_flow
snowfall:
    type: change_context
    properties:
        update_keys:
            intent: 'snowfall'
    transitions:
        flow: weather_flow
tempyesno:

```

```

    type: change_context
    properties:
      update_keys:
        intent: 'tempyesno'
    transitions:
      flow: weather_flow
humidity:
  type: change_context
  properties:
    update_keys:
      intent: 'humidity'
  transitions:
    flow: weather_flow
end_state:
  type: reset
  transitions:
    next_state: init

```

#### Retail , flow.yml

```

name: retail_flow
states:
  store_location:
    type: conditional_exists
    properties:
      key: '{{location}}'
    transitions:
      exists: retail_state
      notexists: ask_retail_location
  retail_state:
    type: Retail
    properties:
    transitions:
      next_state: output_retail
  output_retail:
    type: message_text
    properties:
      text: '{{retail_status}}'
    transitions:
      next_state: input_intent
  ask_retail_location:
    type: message_text
    properties:
      text: "For which location? I can find the closest store near you."
    transitions:

```

```

    next_state: input_retail_location
input_retail_location:
  type: input_user
  properties:
    nlp_type: jarvis
    entities:
      slot: location
    require_match: true
  transitions:
    match: retail_state
    notmatch: check_retail_jarvis_error
output_retail_acknowledge:
  type: message_text_random
  properties:
    responses:
      - 'ok in {{location}}'
      - 'the store in {{location}}'
      - 'I always wanted to shop in {{location}}'
    delay: 0
  transitions:
    next_state: retail_state
output_retail_notlocation:
  type: message_text
  properties:
    text: "I did not understand the location. Can you please repeat?"
  transitions:
    next_state: input_intent
check_rerail_jarvis_error:
  type: conditional_exists
  properties:
    key: '{{jarvis_error}}'
  transitions:
    exists: show_retail_jarvis_api_error
    notexists: output_retail_notlocation
show_retail_jarvis_api_error:
  type: message_text
  properties:
    text: "I am having troubled understanding right now. Come again on that?"
  transitions:
    next_state: input_intent

```

#### weather flow.yml

```
name: weather_flow
```

```

states:
  check_weather_location:
    type: conditional_exists
    properties:
      key: '{{location}}'
    transitions:
      exists: weather_state
      notexists: ask_weather_location
  weather_state:
    type: Weather
    properties:
    transitions:
      next_state: output_weather
  output_weather:
    type: message_text
    properties:
      text: '{{weather_status}}'
    transitions:
      next_state: input_intent
  ask_weather_location:
    type: message_text
    properties:
      text: "For which location?"
    transitions:
      next_state: input_weather_location
  input_weather_location:
    type: input_user
    properties:
      nlp_type: jarvis
      entities:
        slot: location
        require_match: true
    transitions:
      match: weather_state
      notmatch: check_jarvis_error
  output_weather_acknowledge:
    type: message_text_random
    properties:
      responses:
        - 'ok in {{location}}'
        - 'the weather in {{location}}'
        - 'I always wanted to go in {{location}}'
      delay: 0
    transitions:
      next_state: weather_state
  output_weather_notlocation:

```

```

    type: message_text
    properties:
      text: "I did not understand the location, can you please repeat?"
    transitions:
      next_state: input_intent
  check_jarvis_error:
    type: conditional_exists
    properties:
      key: '{{jarvis_error}}'
    transitions:
      exists: show_jarvis_api_error
      notexists: output_weather_notlocation
  show_jarvis_api_error:
    type: message_text
    properties:
      text: "I am having troubled understanding right now. Come again on that, else check jarvis services?"
    transitions:
      next_state: input_intent

```

#### error\_flow.yml

```

name: error_flow
states:
  error_state:
    type: message_text_random
    properties:
      responses:
        - "Sorry I didn't get that!"
        - "Are we talking about retail or weather? What would you like to know?"
        - "Sorry I know only about retail information or the weather"
        - "You can ask me about retail information or the weather, the rainfall, the temperature. I don't know much more"
        - "Let's talk about retail or the weather!"
      delay: 0
    transitions:
      next_state: input_intent

```

以履行引擎的形式连接到第三方 **API**

我们将以下第三方 API 作为履行引擎连接到问题解答问题：

- ["WeatherStack API"](#)：返回给定位置的天气，温度，雨和雪。



- ["Yelp Fusion API"](#): 返回给定位置中最接近的存储信息。
- ["eBay Python SDK"](#): 返回给定项目的价格。

## NetApp 零售助理演示

我们录制了 NetApp 零售助理（Nara）的演示视频。

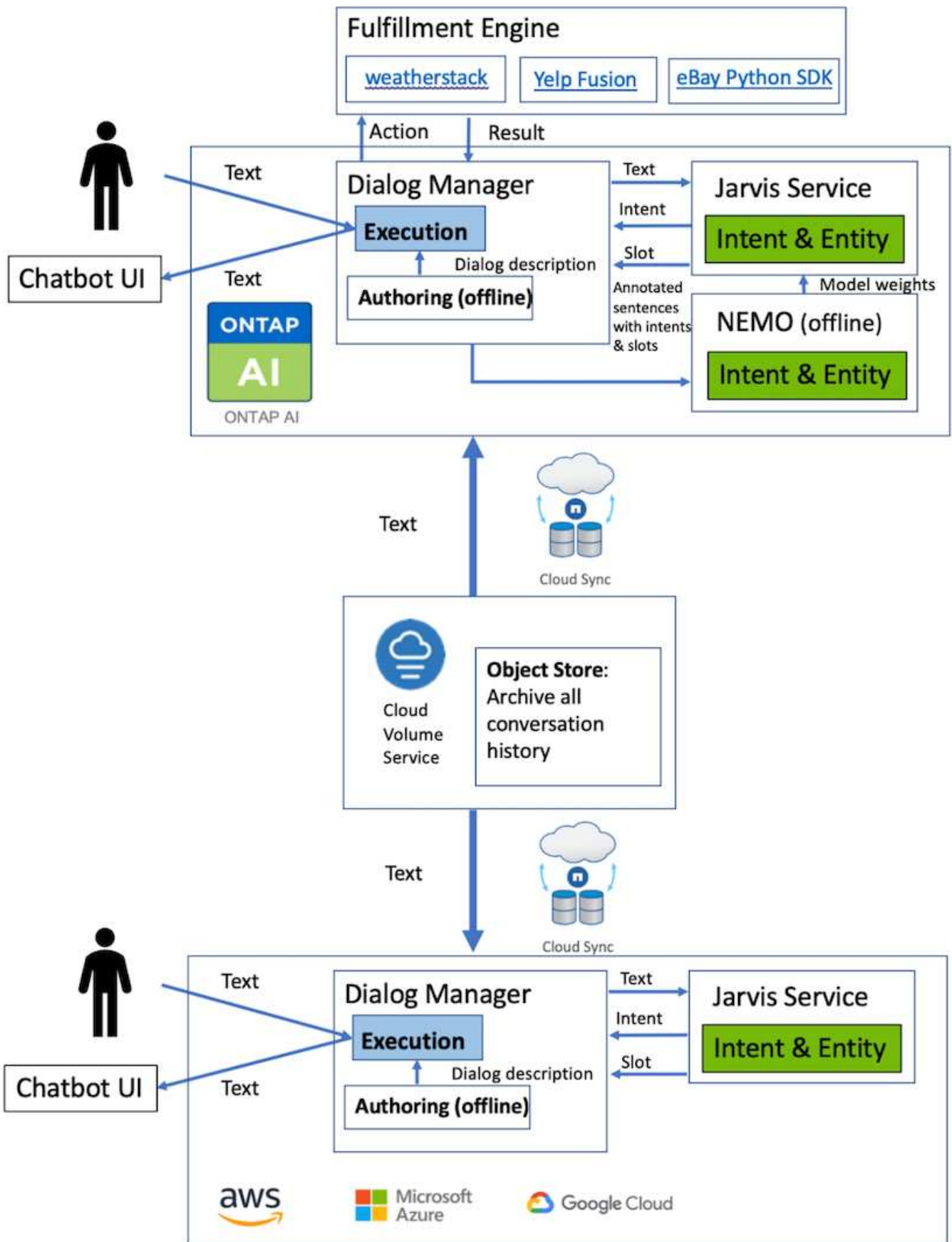
视频演示的一个例子

[视频演示的一个例子](#)



## 使用NetApp BlueXP复制和同步以存档对话历史记录

通过每天将对话历史记录转储到CSV文件一次、我们可以利用BlueXP Copy and Sync将日志文件下载到本地存储。下图显示了在内部和公有云中部署Jarvis、同时使用BlueXP Copy and Sync为Nemo培训发送对话历史记录的架构。有关 Nemo 培训的详细信息，请参见一节 ["使用 Nemo 培训扩展意向模型"](#)。



NVIDIA Nemo 是由 NVIDIA 构建的一个工具包，用于创建对话式 AI 应用程序。该工具包包含一系列针对 ASL，NLP 和 TTS- 的预培训模块，使研究人员和数据科学家能够轻松构建复杂的神经网络架构，并更加专注于设计自己的应用程序。

如上例所示，Nara 只能处理有限类型的问题。这是因为经过预先培训的 NLP 模型只会对这些类型的问题进行训练。如果我们希望 Nara 能够处理更广泛的问题，我们需要使用自己的数据集对其进行重新训练。因此，我们将在此演示如何使用 Nemo 扩展 NLP 模型以满足要求。我们首先将从 Nara 收集的日志转换为 Nemo 的格式，然后训练数据集以增强 NLP 模型。

### 型号

我们的目标是使 Nara 能够根据用户首选项对项目进行排序。例如，我们可能会要求 Nara 推荐排名最高的寿司店，也可能希望 Nara 寻找价格最低的 jeans。为此，我们使用 Nemo 中提供的意向检测和插槽填充模型作为我们的培训模型。通过此模型，Nara 可以了解搜索首选项的意图。

### 数据准备

为了训练模型，我们会收集此类问题的数据集，并将其转换为 Nemo 格式。我们在此处列出了用于训练模型的文件。

#### dict.intents.csv

此文件列出了我们希望 Nemo 了解的所有意向。此处，我们有两个主要意向，一个意图仅用于对不符合任何主要意向的问题进行分类。

```
price_check
find_the_store
unknown
```

#### dict.slots.csv

此文件列出了我们可以在培训问题上标记的所有插槽。

```
B-store.type
B-store.name
B-store.status
B-store.hour.start
B-store.hour.end
B-store.hour.day
B-item.type
B-item.name
B-item.color
B-item.size
B-item.quantity
B-location
B-cost.high
```

```
B-cost.average
B-cost.low
B-time.period_of_time
B-rating.high
B-rating.average
B-rating.low
B-interrogative.location
B-interrogative.manner
B-interrogative.time
B-interrogative.personal
B-interrogative
B-verb
B-article
I-store.type
I-store.name
I-store.status
I-store.hour.start
I-store.hour.end
I-store.hour.day
I-item.type
I-item.name
I-item.color
I-item.size
I-item.quantity
I-location
I-cost.high
I-cost.average
I-cost.low
I-time.period_of_time
I-rating.high
I-rating.average
I-rating.low
I-interrogative.location
I-interrogative.manner
I-interrogative.time
I-interrogative.personal
I-interrogative
I-verb
I-article
O
```

### Traine.tsv

这是主要的培训数据集。每行都以文件 dict.intent.csv 中列出的意图类别后面的问题开头。此标签将从零开始枚举。

## Train\_slots.tsv

```
20 46 24 25 6 32 6
52 52 24 6
23 52 14 40 52 25 6 32 6
...
```

### 训练模型

```
docker pull nvcr.io/nvidia/nemo:v0.10
```

然后，我们将使用以下命令启动此容器。在此命令中，我们会将容器限制为使用单个 GPU（GPU ID = 1），因为这是一项轻型训练练习。此外，我们还会将本地工作空间 /workspace/nemo/ 映射到容器 /nemo 中的文件夹。

```
NV_GPU='1' docker run --runtime=nvidia -it --shm-size=16g \
    --network=host --ulimit memlock=-1 --ulimit
stack=67108864 \
    -v /workspace/nemo:/nemo\
    --rm nvcr.io/nvidia/nemo:v0.10
```

在容器中，如果要从最初的预先培训的 Bert 模型开始，我们可以使用以下命令启动培训操作步骤。data\_dir 是用于设置训练数据路径的参数。work\_dir 用于配置检查点文件的存储位置。

```
cd examples/nlp/intent_detection_slot_tagging/
python joint_intent_slot_with_bert.py \
    --data_dir /nemo/training_data\
    --work_dir /nemo/log
```

如果我们有新的培训数据集并希望改进先前的模型，则可以使用以下命令从停止的位置继续操作。checkpoint\_dir 获取上一个检查点文件夹的路径。

```
cd examples/nlp/intent_detection_slot_tagging/
python joint_intent_slot_infer.py \
    --data_dir /nemo/training_data \
    --checkpoint_dir /nemo/log/2020-05-04_18-34-20/checkpoints/ \
    --eval_file_prefix test
```

### 推理模型

我们需要在经过一定次数的时间之后验证经过训练的模型的性能。使用以下命令，我们可以逐个测试查询。例如，在此命令中，我们希望检查我们的模型是否能够正确识别查询的目的 在哪里可以获得最好的意大利面。

```
cd examples/nlp/intent_detection_slot_tagging/
python joint_intent_slot_infer_b1.py \
--checkpoint_dir /nemo/log/2020-05-29_23-50-58/checkpoints/ \
--query "where can i get the best pasta" \
--data_dir /nemo/training_data/ \
--num_epochs=50
```

然后，以下是推理的输出。在输出中，我们可以看到经过培训的模型可以正确预测 `DETAINT_the_store` 的意向，并返回我们感兴趣的关键字。通过这些关键字，我们可以使 Nara 搜索用户所需内容并进行更精确的搜索。

```
[NeMo I 2020-05-30 00:06:54 actions:728] Evaluating batch 0 out of 1
[NeMo I 2020-05-30 00:06:55 inference_utils:34] Query: where can i get the
best pasta
[NeMo I 2020-05-30 00:06:55 inference_utils:36] Predicted intent:      1
find_the_store
[NeMo I 2020-05-30 00:06:55 inference_utils:50] where      B-
interrogative.location
[NeMo I 2020-05-30 00:06:55 inference_utils:50] can        O
[NeMo I 2020-05-30 00:06:55 inference_utils:50] i          O
[NeMo I 2020-05-30 00:06:55 inference_utils:50] get        B-verb
[NeMo I 2020-05-30 00:06:55 inference_utils:50] the        B-article
[NeMo I 2020-05-30 00:06:55 inference_utils:50] best       B-rating.high
[NeMo I 2020-05-30 00:06:55 inference_utils:50] pasta      B-item.type
```

## 结论

一个真正的人工智能对话系统可以进行类似于人类的对话，了解相关背景并提供智能响应。此类 AI 模型通常非常庞大且非常复杂。借助 NVIDIA GPU 和 NetApp 存储，可以对大规模的一流语言模型进行培训和优化，以快速运行推理。这是一个重大的进步，旨在结束快速 AI 模型与大型复杂 AI 模型之间的权衡。GPU 优化的语言理解模式可以集成到医疗保健，零售和金融服务等行业的 AI 应用程序中，为智能扬声器和客户服务线中的高级数字语音助理提供支持。通过这些高质量的对话式 AI 系统，各个垂直行业的企业可以在与客户接洽时提供以前无法实现的个性化服务。

借助 JARVIS，可以部署虚拟助手，数字 avatars，多模式传感器 Fusion（CV 与 ASS/NLP/TTs 融合）或任何 ASS/NLP/TTS/CV 独立用例，如记录。我们构建了一个虚拟零售助理，可以对天气，关注点和库存定价方面的问题进行问题解答处理。我们还演示了如何通过使用 BlueXP Copy and Sync 归档对话历史记录并对 Nemo 模型进行新数据训练来提高对话 AI 系统的自然语言理解能力。

## 致谢

作者衷心感谢我们尊敬的 NVIDIA 同事：达维德·奥诺弗罗，Alex Qi，Sicong Ji，Marty Jain 和 Robert Sohigian 为本白皮书所做的贡献。作者还想感谢 NetApp 团队的主要成员所做的贡献：Santosh Rao，David Arnette，Michael Oglesby，Brent Davis，

Andy Sayare , Erik Minder 和 Mike McNamara 。

我们衷心感谢所有这些人士，他们提供了洞察力和专业知识，为本白皮书的编写提供了极大的帮助。

## 从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请参见以下资源：

- NVIDIA DGX Station , V100 GPU , GPU Cloud
  - NVIDIA DGX Station<https://www.nvidia.com/en-us/data-center/dgx-station/><sup>[1]</sup>
  - NVIDIA V100 Tensor 核心 GPU<https://www.nvidia.com/en-us/data-center/tesla-v100/><sup>[2]</sup>
  - NVIDIA NGC<https://www.nvidia.com/en-us/gpu-cloud/><sup>[3]</sup>
- NVIDIA JarVis 多模式框架
  - NVIDIA JarVis<https://developer.nvidia.com/nvidia-jarvis><sup>[4]</sup>
  - NVIDIA JarVis Early Access<https://developer.nvidia.com/nvidia-jarvis-early-access><sup>[5]</sup>
- NVIDIA Nemo
  - NVIDIA Nemo<https://developer.nvidia.com/nvidia-nemo><sup>[6]</sup>
  - 开发人员指南<https://nvidia.github.io/NeMo/><sup>[7]</sup>
- NetApp AFF 系统
  - NetApp AFF A 系列产品规格<https://www.netapp.com/us/media/ds-3582.pdf><sup>[8]</sup>
  - 适用于全闪存 FAS 的 NetApp 闪存优势<https://www.netapp.com/us/media/ds-3733.pdf><sup>[9]</sup>
  - ONTAP 9 信息库<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286><sup>[10]</sup>
  - NetApp ONTAP FlexGroup Volumes 技术报告<https://www.netapp.com/us/media/tr-4557.pdf><sup>[11]</sup>
- NetApp ONTAP AI
  - 采用 DGX-1 的 ONTAP AI 和 Cisco 网络设计指南<https://www.netapp.com/us/media/nva-1121-design.pdf><sup>[12]</sup>
  - 《采用 DGX-1 的 ONTAP AI 和 Cisco 网络部署指南》<https://www.netapp.com/us/media/nva-1121-deploy.pdf><sup>[13]</sup>
  - 采用 DGX-1 和 Mellanox 网络设计指南的 ONTAP AI<http://www.netapp.com/us/media/nva-1138-design.pdf><sup>[14]</sup>
  - 采用 DGX-2 的 ONTAP AI 设计指南<https://www.netapp.com/us/media/nva-1135-design.pdf><sup>[15]</sup>



# TR-4858：《NetApp Orchestration 解决方案与 Run：AI》

Rick Huang，David Arnette，Sung-Han Lin，NetApp Yaron Goldberg，Run：AI

NetApp AFF 存储系统可提供极致性能和行业领先的混合云数据管理功能。NetApp 与 Run：AI 合作展示了 NetApp ONTAP AI 解决方案在人工智能（AI）和机器学习（ML）工作负载方面的独特功能，这些功能可提供企业级性能，可靠性和支持。Run：AI 工作负载的 AI 流程编排增加了一个基于 Kubernetes 的计划和资源利用率平台，可帮助研究人员管理和优化 GPU 利用率。NetApp，NVIDIA 和 Run：AI 的解决方案与 NVIDIA DGX 系统相结合，可提供一个专为企业 AI 工作负载构建的基础架构堆栈。本技术报告为客户构建对话式 AI 系统以支持各种用例和行业垂直市场提供了方向性指导。其中包括有关部署 Run：AI 和 NetApp AFF A800 存储系统的信息，并可作为参考架构来快速成功地部署 AI 计划。

解决方案的目标受众包括以下组：

- 为容器化微服务等基于 Kubernetes 的用例设计 AI 模型和软件开发解决方案的企业架构师
- 数据科学家正在寻找高效的方法，以便在具有多个团队和项目的集群环境中实现高效的模型开发目标
- 负责维护和运行生产模型的数据工程师
- 希望打造最佳 Kubernetes 集群资源利用体验并通过 AI 计划缩短上市时间的高管和 IT 决策者和业务主管

## 解决方案概述

本节提供了适用于 ONTAP AI 的 Run：AI 解决方案的解决方案概述。

### NetApp ONTAP AI 和 AI 控制平台

由 NetApp 和 NVIDIA 开发并验证的 NetApp ONTAP AI 架构由 NVIDIA DGX 系统和 NetApp 云连接存储系统提供支持。此参考架构为 IT 组织提供了以下优势：

- 消除设计复杂性
- 支持独立扩展计算和存储
- 支持客户从小规模入手，无缝扩展
- 为各种性能和成本点提供了一系列存储选项

NetApp ONTAP AI 将 DGX 系统和 NetApp AFF A800 存储系统与一流的网络紧密集成在一起。NetApp ONTAP AI 和 DGX 系统消除了设计复杂性和猜测性工作，从而简化了 AI 部署。客户可以从小规模入手，无中断地扩展系统，同时智能地管理从边缘到核心再到云再到云的数据。

NetApp AI 控制平台是一个全堆栈 AI，ML 和深度学习（DL）数据和实验管理解决方案，适用于数据科学家和数据工程师。随着企业越来越多地使用 AI，他们面临着许多挑战，包括工作负载可扩展性和数据可用性。NetApp AI 控制平台可通过多种功能来应对这些挑战，例如像 Git repo 一样快速克隆数据命名空间，以及定义和实施 AI 培训工作流，这些工作流可以近乎即时地创建数据和模型基线，以实现可追溯性和版本控制。借助 NetApp AI 控制平台，您可以在站点和区域之间无缝复制数据，并快速配置 Jupyter 笔记本工作空间，以便访问海量数据集。

运行：适用于 AI 工作负载编排的 AI 平台

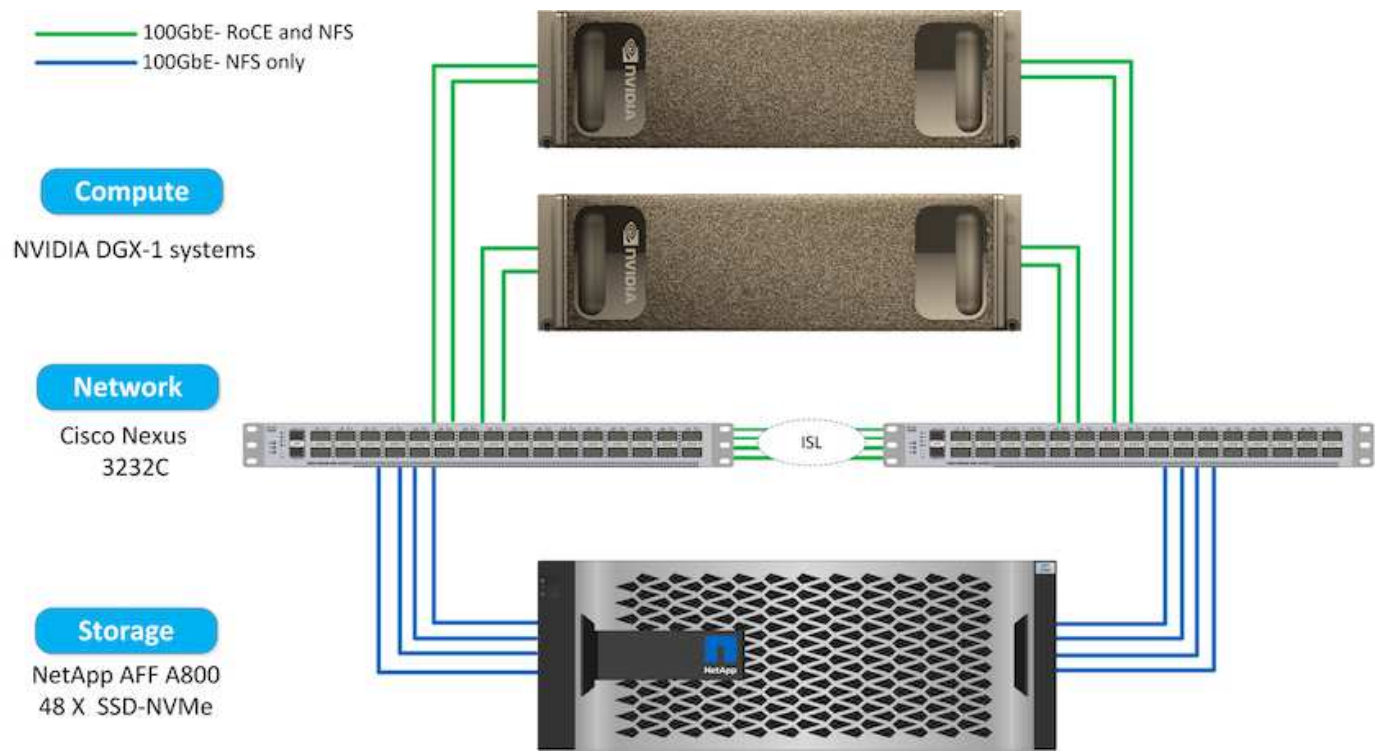
运行：AI 为 AI 基础架构构建了全球首款业务流程和虚拟化平台。通过将工作负载从底层硬件中抽象出来，Run：AI 可创建一个可动态配置的 GPU 资源共享池，从而高效地编排 AI 工作负载并优化 GPU 的使用。数据科学家可以无缝地使用大量 GPU 功能来改进和加快研究速度，同时 IT 团队可以对资源配置，队列和利用率保持集中的跨站点控制和实时可见性。Run：AI 平台基于 Kubernetes 构建，可与现有 IT 和数据科学工作流程轻松集成。

Run：AI 平台具有以下优势：

- \* 加快创新速度。\* 通过将 Run：AI 资源池，队列和优先级划分机制与 NetApp 存储系统结合使用，研究人员可以从基础架构管理的麻烦中消除，并可以专注于数据科学。运行：AI 和 NetApp 客户可以根据需要运行任意数量的工作负载，而不会出现计算或数据管道瓶颈，从而提高工作效率。
- \* 提高团队工作效率。\* 运行：AI 公平算法可确保所有用户和团队都能获得公平的资源份额。可以预设优先级项目的策略，该平台支持将资源从一个用户或团队动态分配给另一个用户或团队，从而帮助用户及时访问所需的 GPU 资源。
- \* 提高了 GPU 利用率。\* 运行：AI 计划程序使用户能够轻松地使用百分比 GPU，整数 GPU 和多个 GPU 节点在 Kubernetes 上进行分布式培训。这样，AI 工作负载就可以根据您的需求运行，而不是根据容量运行。数据科学团队可以在同一基础架构上运行更多 AI 实验。

解决方案技术

此解决方案是在一个 NetApp AFF A800 系统，两个 DGX-1 服务器和两个 Cisco Nexus 3232C 100GbE 交换机上实施的。每个 DGX-1 服务器都通过四个 100GbE 连接连接到 Nexus 交换机，这些连接可通过使用基于融合以太网（RoCE）的远程直接内存访问（RDMA）进行 GPU 间通信。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四个 100GbE 链路连接到网络交换机。下图显示了本技术报告中用于所有测试场景的 ONTAP AI 解决方案架构。



此解决方案中使用的硬件

此解决方案已使用 ONTAP AI 参考架构两个 DGX-1 节点和一个 AFF A800 存储系统进行了验证。请参见 ["NVA-1121"](#) 有关此验证中使用的基础架构的更多详细信息。

下表列出了在测试中实施解决方案所需的硬件组件。

硬件	数量
DGX-1 系统	2.
AFF A800	1.
Nexus 3232C 交换机	2.

软件要求

此解决方案已通过安装了 Run ： AI 操作员的基本 Kubernetes 部署进行验证。Kubernetes 是使用部署的 ["NVIDIA DeepOps"](#) 部署引擎，用于为生产就绪环境部署所有必需的组件。DeepOps 会自动部署 ["NetApp Trident"](#) 为了与 K8s 环境实现持久存储集成，我们创建了默认存储类，以便容器可以利用 AFF A800 存储系统中的存储。有关在 ONTAP AI 上使用 Kubernetes 进行 Trident 的详细信息，请参见 ["TR-4798"](#)。

下表列出了在测试中实施解决方案所需的软件组件。

软件	版本或其他信息
NetApp ONTAP 数据管理软件	9.6 第 4 页
Cisco NX-OS 交换机固件	7.0 （ 3 ） I6 （ 1 ）
NVIDIA DGX 操作系统	4.0.4 — Ubuntu 18.04 LTS
Kubernetes 版本	1.17
Trident 版本	20.04.0
运行： AI 命令行界面	v2.1.13
运行： AI Orchestration Kubernetes Operator 版本	1.0.39
Docker 容器平台	18.06.1-ce [e68fc7a]

有关 Run ： AI 的其他软件要求，请参见 ["运行： AI GPU 集群前提条件"](#)。

使用 **Run ： AI** 优化集群和 **GPU** 利用率

以下各节详细介绍了运行： AI 安装，测试场景以及在此验证中执行的结果。

我们使用行业标准基准测试工具（包括 TensorFlow 基准测试）验证了此系统的运行和性能。ImageNet 数据集用于训练 RESNET-50 ，它是一种著名的神经网络（ Convolutional Neural Network ， CNN ） DL 图像分类模型。RESNET-50 可提供准确的训练结果，并缩短处理时间，从而使我们能够对存储产生足够的需求。

运行： **AI** 安装

要安装 Run ： AI ，请完成以下步骤：

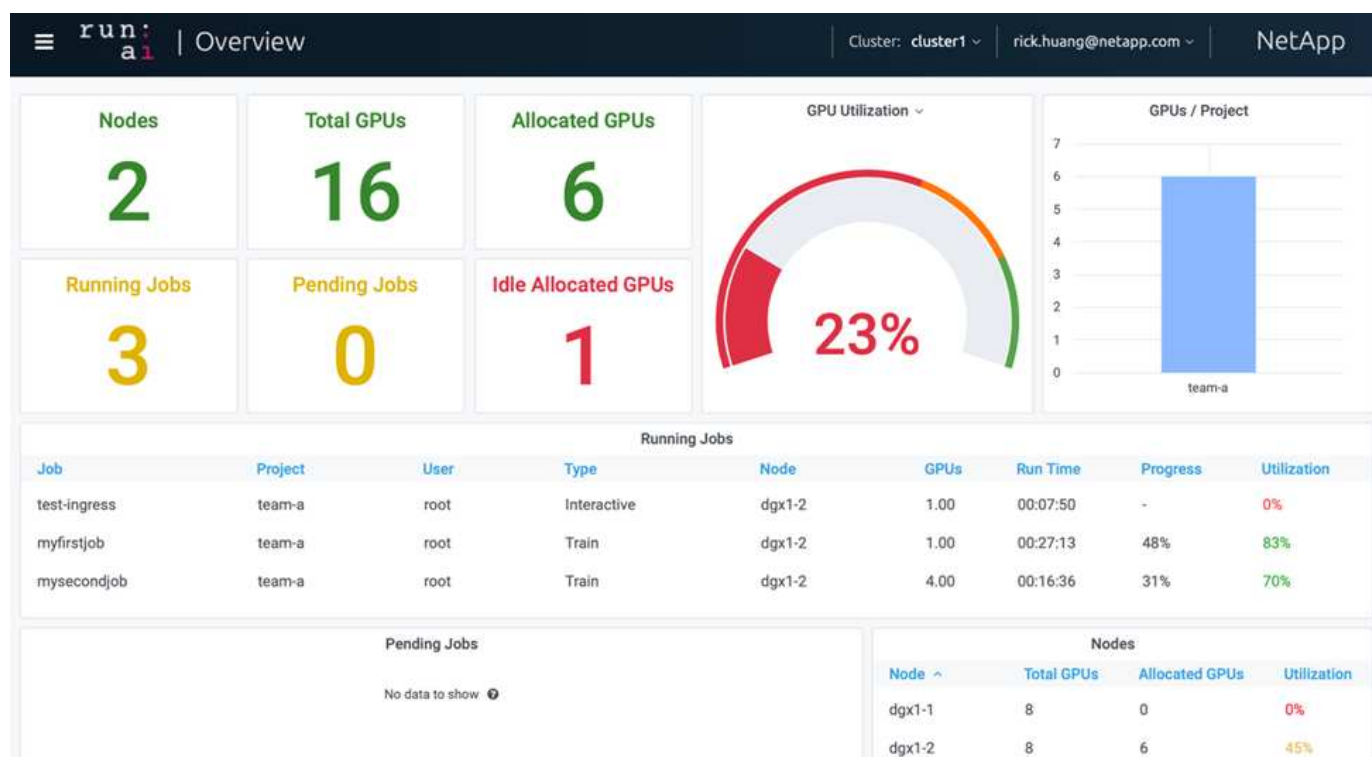
1. 使用 DeepOps 安装 Kubernetes 集群并配置 NetApp 默认存储类。
2. 准备 GPU 节点：
  - a. 验证是否已在 GPU 节点上安装 NVIDIA 驱动程序。
  - b. 验证是否已安装 nvidia-Docker 并将其配置为默认 Docker 运行时。
3. 安装运行：AI：
  - a. 登录到 "运行：AI 管理员 UI" 以创建集群。
  - b. 下载创建的 runai-operator-<clustername>.yaml 文件。
  - c. 将操作员配置应用于 Kubernetes 集群。

```
kubectl apply -f runai-operator-<clustername>.yaml
```

4. 验证安装。
  - a. 转至 "<https://app.run.ai/>"。
  - b. 转到 "概述" 信息板。
  - c. 验证右上角的 GPU 数量是否反映了预期的 GPU 数量，并且 GPU 节点均位于服务器列表中。有关 Run：AI 部署的详细信息，请参见 "[在内部 Kubernetes 集群上安装 Run：AI](#)" 和 "[安装 Run：AI 命令行界面](#)"。

#### 运行：AI 信息板和视图

在 Kubernetes 集群上安装 Run：AI 并正确配置容器后，您将看到以下信息板和视图 "<https://app.run.ai/>" 在浏览器中，如下图所示。



集群中共有 16 个 GPU，由两个 DGX-1 节点提供。您可以查看节点数，可用 GPU 总数，分配给工作负载的已分配 GPU，正在运行的作业总数，待定作业以及闲置已分配 GPU。右侧的条形图显示每个项目的 GPU，其中总结了不同团队如何使用集群资源。中间是当前正在运行的作业列表，其中包含作业详细信息，包括作业名称，项目，用户，作业类型，每个作业正在运行的节点，为此作业分配的 GPU 数量，作业的当前运行时间，作业进度百分比以及该作业的 GPU 利用率。请注意，集群利用率不足（GPU 利用率为 23%），因为一个团队只提交了三个正在运行的作业（team-A）。

在下一节中，我们将介绍如何在“项目”选项卡中创建多个团队，并为每个团队分配 GPU，以便在每个集群有大量用户时最大限度地提高集群利用率并管理资源。这些测试场景模拟了在训练，推理和交互式工作负载之间共享内存和 GPU 资源的企业环境。

## 为数据科学团队创建项目并分配 GPU

研究人员可以通过 Run：AI CLI，Kubeflow 或类似流程提交工作负载。为了简化资源分配并创建优先级，Run：AI 引入了项目概念。项目是指将项目名称与 GPU 分配和首选项关联的配额实体。这是一种管理多个数据科学团队的简单便捷的方式。

提交工作负载的研究人员必须将项目与工作负载请求相关联。运行：AI 计划程序会将请求与当前分配和项目进行比较，并确定是否可以工作负载分配资源或是否应保持待定状态。

作为系统管理员，您可以在 Run：AI projects 选项卡中设置以下参数：

- \* 模拟项目。\* 为每个用户设置一个项目，为每个用户团队设置一个项目，并为每个实际组织项目设置一个项目。
- \* 项目配额。\* 每个项目都与一个 GPU 配额相关联，可以同时为此项目分配 GPU 配额。这是一个有保障的配额，因为使用此项目的研究人员无论在集群中处于何种状态，都可以获得这一数量的 GPU。通常，项目分配总和应等于集群中的 GPU 数量。除此之外，此项目的用户还可以收到超配额。只要未使用 GPU，使用此项目的研究人员就可以获得更多 GPU。我们在中演示了超额配额测试场景和公平考虑事项 ["通过过度配额 GPU 分配实现高集群利用率"](#)，["基本资源分配公平"](#)，和 ["配额过度公平"](#)。
- 创建新项目，更新现有项目并删除现有项目。
- \* 限制作业在特定节点组上运行 \*。您可以分配仅在特定节点上运行的特定项目。如果项目团队需要专用硬件，例如具有足够内存的硬件，则此功能非常有用。或者，项目团队也可能是特定硬件的所有者，这些硬件是通过专门预算获得的，或者您可能需要直接构建或交互式工作负载来处理较弱的硬件，并将较长的培训或无人看管的工作负载定向到较快的节点。有关对节点进行分组并为特定项目设置相关性的命令，请参见 ["运行：AI 文档"](#)。
- \* 限制交互作业的持续时间 \*。研究人员经常忘记关闭交互式作业。这可能会导致资源浪费。一些组织倾向于限制交互式作业的持续时间并自动关闭这些作业。

下图显示了已创建四个团队的“项目”视图。每个团队分配不同数量的 GPU 来处理不同的工作负载，GPU 总数等于由两个 DGX-1 组成的集群中可用 GPU 总数。

run:ai

Projects





Cluster: cluster1

rick.huang@netapp.com

NetApp

Filter and Search

+ Add New project

Project Name	Assigned GPUs	Created	Training Node Affinity	Interactive Node Affinity
 team-a	2	07/27/20, 9:28AM	none	none
 team-b	4	07/28/20, 7:50AM	none	none
 team-c	2	07/28/20, 7:50AM	none	none
 team-d	8	07/28/20, 7:51AM	none	none

在 **Run : AI** 命令行界面中提交作业

本节详细介绍了可用于运行任何 Kubernetes 作业的基本 **Run : AI** 命令。它会根据工作负载类型分为三部分。AI/ML/DL 工作负载可分为两种通用类型：

- \* 无人参与的培训课程 \*。对于这些类型的工作负载，数据科学家会准备一个自运行的工作负载并将其发送给执行。执行期间，客户可以检查结果。此类工作负载通常用于生产或模型开发阶段，无需人工干预。
- \* 交互式构建会话 \*。对于这些类型的工作负载，数据科学家将与 Bash ， Jupyter Notebook ， 远程 PyCharm 或类似的 IDE 打开交互式会话，并直接访问 GPU 资源。我们还提供了第三种使用连接的端口运行交互式工作负载的方案，以便向容器用户显示内部端口。

无人参与的培训工作负载

设置项目并分配 GPU 后，您可以在命令行中使用以下命令运行任何 Kubernetes 工作负载：

```
$ runai project set team-a runai submit hyper1 -i gcr.io/run-ai-demo/quickstart -g 1
```

此命令将为团队 A 启动无人参与的培训作业，并分配一个 GPU 。此作业基于示例 Docker 映像 `gcr.io/run-ai-demo/Quickstart` 。我们将作业命名为 `hyper1` 。然后，您可以运行以下命令来监控作业的进度：

```
$ runai list
```

下图显示了 `runai list` 命令的结果。您可能看到的典型状态包括：

- 容器创建 。正在从云存储库下载 Docker 容器。
- 待定 。作业正在等待计划。
- 运行 。作业正在运行。

```

You can run 'runai get hyper1 -p team-a' to check the job status
~> runai list
Showing jobs for project team-a
NAME      STATUS  AGE  NODE                                     IMAGE                                     TYPE      PROJECT  USER  GPUs
hyper1    Running  11s  gke-dev-yaron1-gpu-4-pool-154f511d-5nk5 gcr.io/run-ai-demo/quickstart          Train    team-a   yaron   1

```



要获取作业的其他状态，请运行以下命令：

```
$ runai get hyper1
```

要查看作业日志，请运行 `runai logs <job-name>` 命令：

```
$ runai logs hyper1
```

在此示例中，您应看到正在运行的 DL 会话的日志，包括当前训练时间，ETA，损失函数值，准确性以及每个步骤所用时间。

您可以在运行：AI UI 上查看集群状态，网址为 "<https://app.run.ai/>"。在 Dashboards > Overview 下，您可以监控 GPU 利用率。

要停止此工作负载，请运行以下命令：

```
$ runai delete hyper1
```

此命令将停止训练工作负载。您可以再次运行 `runai list` 来验证此操作。有关详细信息，请参见 "[启动无人参与的培训工作负载](#)"。

#### 交互式构建工作负载

设置项目并分配 GPU 后，您可以在命令行中使用以下命令运行交互式构建工作负载：

```
$ runai submit build1 -i python -g 1 --interactive --command sleep --args infinity
```

此作业基于示例 Docker 映像 `python`。我们将作业 BUILD1 命名为。



`- 交互式` 标志表示作业没有开始或结束研究人员有责任完成此项工作。管理员可以为交互式作业定义一个时间限制，在该时间限制之后，系统会终止这些作业。

`-g 1` 标志可为此作业分配一个 GPU。提供的命令和参数为 `—命令休眠— args infinity`。您必须提供命令，否则容器将立即启动并退出。

以下命令的工作方式与中所述的命令类似 [\[无人参与的培训工作负载\]](#)：

- `runai list`：显示名称，状态，期限，节点，映像，用于作业的项目，用户和 GPU。
- `runai get build1`：显示作业 `build1` 的其他状态。
- `runai delete build1`：停止交互式工作负载 BUILD1。To get a bash shell to the container， the following command：

```
$ runai bash build1
```

这样就可以直接将 shell 连接到计算机。然后，数据科学家可以在容器中开发或微调其模型。

您可以在运行：AI UI 上查看集群状态，网址为 "<https://app.run.ai>"。有关详细信息，请参见 "[启动和使用交互式构建工作负载](#)"。

#### 使用已连接端口的交互式工作负载

作为交互式构建工作负载的扩展，在使用 Run：AI CLI 启动容器时，您可以向容器用户显示内部端口。这对于云环境，使用 Jupyter 笔记本电脑或连接到其他微服务非常有用。"传入" 允许从 Kubernetes 集群外部访问 Kubernetes 服务。您可以通过创建一组规则来配置访问，这些规则定义哪些入站连接访问哪些服务。

为了更好地管理对集群中服务的外部访问，我们建议集群管理员安装 "传入" 并配置负载均衡器。

要使用传入作为服务类型，请在提交工作负载时运行以下命令以设置方法类型和端口：

```
$ runai submit test-ingress -i jupyter/base-notebook -g 1 \
--interactive --service-type=ingress --port 8888 \
--args="--NotebookApp.base_url=test-ingress" --command=start-notebook.sh
```

容器成功启动后、执行 `runai list` 以查看 SERVICE URL(S) 访问 Jupyter 笔记本。此 URL 由入口端点，作业名称和端口组成。

有关详细信息，请参见 "[使用连接的端口启动交互式构建工作负载](#)"。

#### 实现高集群利用率

在本节中，我们模拟了一个实际场景，其中四个数据科学团队各自提交自己的工作负载，以展示 Run：AI Orchestration 解决方案，它可以在保持 GPU 资源优先级和平衡的同时实现高集群利用率。我们首先使用一节中所述的 RESNET-50 基准测试 "[使用 ImageNet 数据集的 RESNET-50 基准测试摘要](#)"：

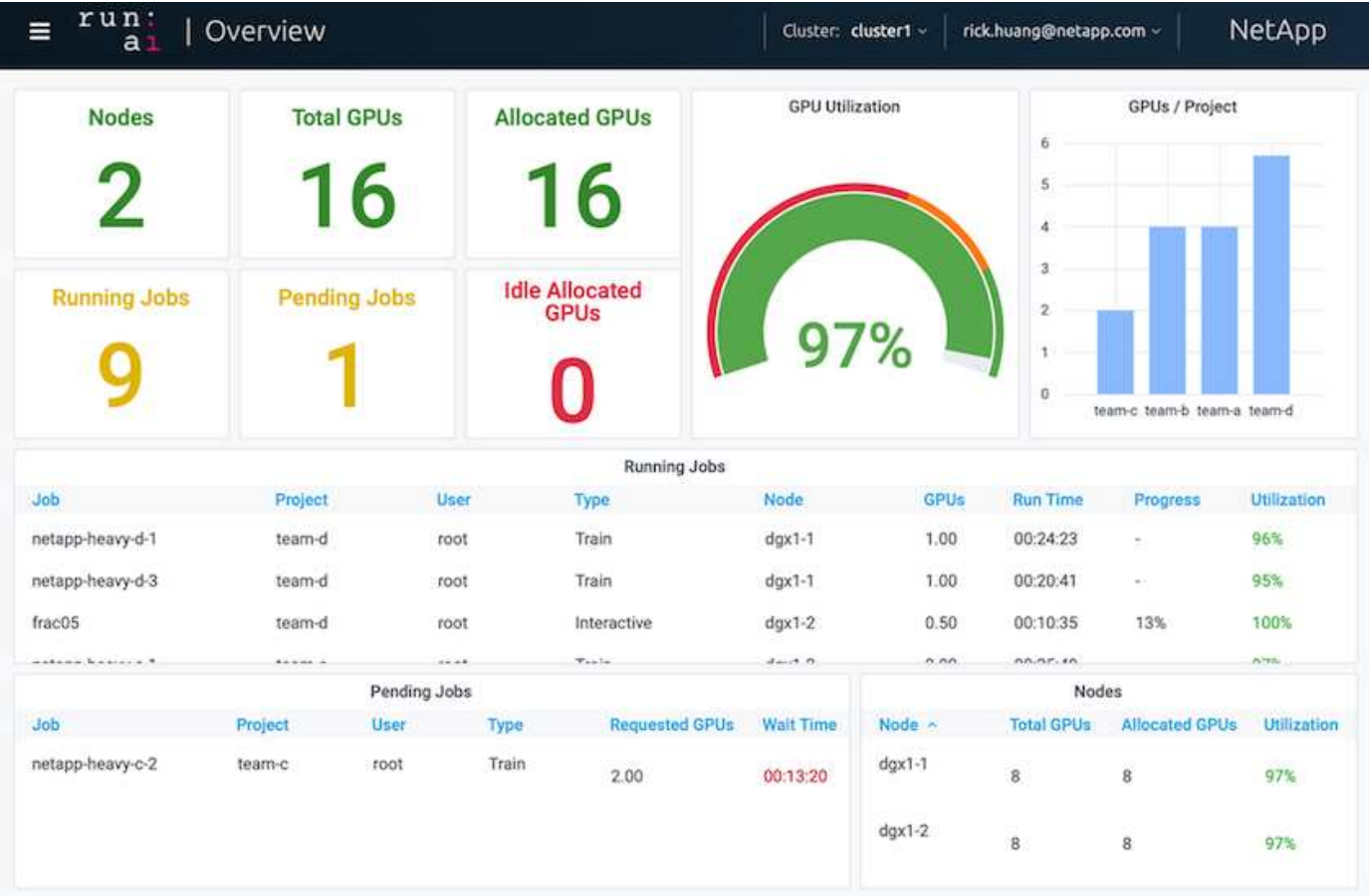
```
$ runai submit netapp1 -i netapp/tensorflow-tf1-py3:20.01.0 --local-image
--large-shm -v /mnt:/mnt -v /tmp:/tmp --command python --args
"/netapp/scripts/run.py" --args "--
dataset_dir=/mnt/mount_0/dataset/imagenet/imagenet_original/" --args "--
num_mounts=2" --args "--dgx_version=dgx1" --args "--num_devices=1" -g 1
```

我们运行的 RESNET-50 基准测试与中相同 "[NVA-1121](#)"。对于不驻留在公有 Docker 存储库中的容器，我们使用了标志 `-local-image`。我们将主机 DGX-1 节点上的目录 `/mnt` 和 `/tm` 分别挂载到 `/mnt` 和 `/tm`。该数据集位于 NetApp AFFA800 上，并且 `dataset_dir` 参数指向目录。`-num_devices=1` 和 `-g 1` 表示我们为此作业分配一个 GPU。前者是 `run.py` 脚本的参数，而后者是 `runai Submit` 命令的标志。

下图显示了一个系统概述信息板，其中 GPU 利用率为 97%，所有十六个可用 GPU 均已分配。您可以在 GPU/项目条形图中轻松查看为每个团队分配的 GPU 数量。"正在运行的作业" 窗格显示当前正在运行的作业名称，



项目，用户，类型，节点， GPU 已用，运行时间，进度和利用率详细信息。队列中的工作负载列表及其等待时间显示在 "Pending" 作业中。最后，节点框将提供集群中各个 DGX-1 节点的 GPU 编号和利用率。



为要求较低的工作负载或交互式工作负载分配的 GPU 百分比

当研究人员和开发人员在开发，超参数调整或调试阶段使用其模型时，此类工作负载通常所需的计算资源更少。因此，配置百分比 GPU 和内存的效率更高，以便可以将同一 GPU 同时分配给其他工作负载。Run： AI 的业务流程解决方案为 Kubernetes 上的容器化工作负载提供了一个百分比 GPU 共享系统。该系统支持运行 CUDA 程序的工作负载，尤其适用于推理和模型构建等轻型 AI 任务。部分 GPU 系统可以透明地为数据科学和 AI 工程团队提供在一个 GPU 上同时运行多个工作负载的能力。这样，企业就可以在同一硬件上运行更多的工作负载，例如计算机视觉，语音识别和自然语言处理，从而降低成本。

Run： AI 的百分比 GPU 系统可利用自身的内存和计算空间有效地创建虚拟化逻辑 GPU，容器可以使用和访问这些 GPU，就像它们是独立的处理器一样。这样，多个工作负载便可在同一 GPU 上的容器中并排运行，而不会相互干扰。解决方案是透明，简单且可移植的，不需要对容器本身进行更改。

一个典型的使用情形可能会看到在同一个 GPU 上运行两到八个作业，这意味着您可以使用同一个硬件执行八倍的工作。

对于下图中的作业 frac05 属于项目 team-d，我们可以看到分配的 GPU 数量为 0.5。这一点可通过 nvidia-smi 命令进一步验证，该命令显示容器可用的 GPU 内存为 16，255 MB： DGX-1 节点中每个 V100 GPU 32 GB 的一半。

```
root@run-deploy:~# runai bash frac05 -p team-d
root@frac05-0:/workload# nvidia-smi
Tue Jul 28 15:17:03 2020

+-----+
| NVIDIA-SMI 450.51.05      Driver Version: 450.51.05      CUDA Version: 11.0      |
+-----+-----+-----+-----+-----+-----+
| GPU  Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                               |                      |              MIG M. |
+-----+-----+-----+-----+-----+-----+
|   0   Tesla V100-SXM2...    On      | 00000000:07:00.0 Off |                    0 |
| N/A   57C    P0     240W / 300W | 15525MiB / 16255MiB |    100%      Default |
|                               |                      |                  N/A |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes:                                     |
|  GPU   GI    CI          PID    Type   Process name                  GPU Memory |
|          ID    ID                                   |          Usage   |
+-----+-----+-----+-----+-----+-----+
|   0     N/A  N/A         156      C      python3                      15525MiB |
+-----+
```

通过过度配额 GPU 分配实现高集群利用率

在本节和各节中 "基本资源分配公平", 和 "配额过度公平", 我们设计了高级测试方案, 用于演示运行: AI 流程编排功能, 以实现复杂的工作负载管理, 自动抢占式计划和超配额 GPU 配置。我们这样做是为了在 ONTAP AI 环境中实现高集群资源利用率并优化企业级数据科学团队的工作效率。

对于这三个部分, 请设置以下项目和配额:

项目	配额
团队 A	4.
团队 b	2.
团队 c	2.
团队	8.

此外, 我们还会对这三个部分使用以下容器:

- Jupyter 笔记本电脑: `jupyter/base-notebook`
- Run : AI Quickstart : `gcr.io/run-ai-demo/Quickstart`

我们为此测试场景设定了以下目标:

- 展示资源配置的简便性以及如何从用户中提取资源

- 展示用户如何轻松配置 GPU 的小部分和 GPU 的整数
- 展示如果集群中存在可用 GPU，系统如何通过允许团队或用户超过其资源配额来消除计算瓶颈
- 展示如何在运行计算密集型作业（例如 NetApp 容器）时使用 NetApp 解决方案消除数据管道瓶颈
- 显示如何使用系统运行多种类型的容器
  - Jupyter 笔记本电脑
  - 运行：AI 容器
- 集群已满时显示高利用率

有关在测试期间执行的实际命令序列的详细信息，请参见 ["第 4.8 节的测试详细信息"](#)。

提交所有 13 个工作负载后，您可以看到一个容器名称和分配的 GPU 列表，如下图所示。我们有七个培训和六个互动作业，模拟四个数据科学团队，每个团队都有自己的模型运行或开发。对于交互式作业，各个开发人员都在使用 Jupyter 笔记本电脑编写或调试其代码。因此，它适合在不使用过多集群资源的情况下配置 GPU 分段。

```
root@run-deploy:~# runai list -A
```

NAME	STATUS	AGE	NODE	IMAGE	TYPE	PROJECT	USER	GPUS	CREATED BY CLI	SERVICE URL(S)
b-4-gg	Running	2m	dgx1-2	gcr.io/run-ai-demo/quickstart	Train	team-b	root	2	true	
c-5-g	Running	2m	dgx1-2	gcr.io/run-ai-demo/quickstart	Train	team-c	root	1	true	
c-4-gg	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Train	team-c	root	2	true	
b-3-g	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Train	team-b	root	1	true	
c-3-g02	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Interactive	team-c	root	0.2	true	
d-1-gggg	Running	2m	dgx1-2	gcr.io/run-ai-demo/quickstart	Train	team-d	root	4	true	
c-2-g03	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Interactive	team-c	root	0.3	true	
c-1-g05	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Interactive	team-c	root	0.5	true	
a-2-gg	Running	3m	dgx1-1	gcr.io/run-ai-demo/quickstart	Train	team-a	root	2	true	
b-2-g04	Running	3m	dgx1-2	gcr.io/run-ai-demo/quickstart	Interactive	team-b	root	0.4	true	
a-1-g	Running	3m	dgx1-1	gcr.io/run-ai-demo/quickstart	Train	team-a	root	1	true	
b-1-g06	Running	3m	dgx1-2	gcr.io/run-ai-demo/quickstart	Interactive	team-b	root	0.6	true	
a-1-1-jupyter	Running	3m	dgx1-1	jupyter/base-notebook	Interactive	team-a	root	1	true	http://10.61.218.134/a-1-1-jupyter, https://10.61.218.134/a-1-1-jupyter

此测试场景的结果如下：

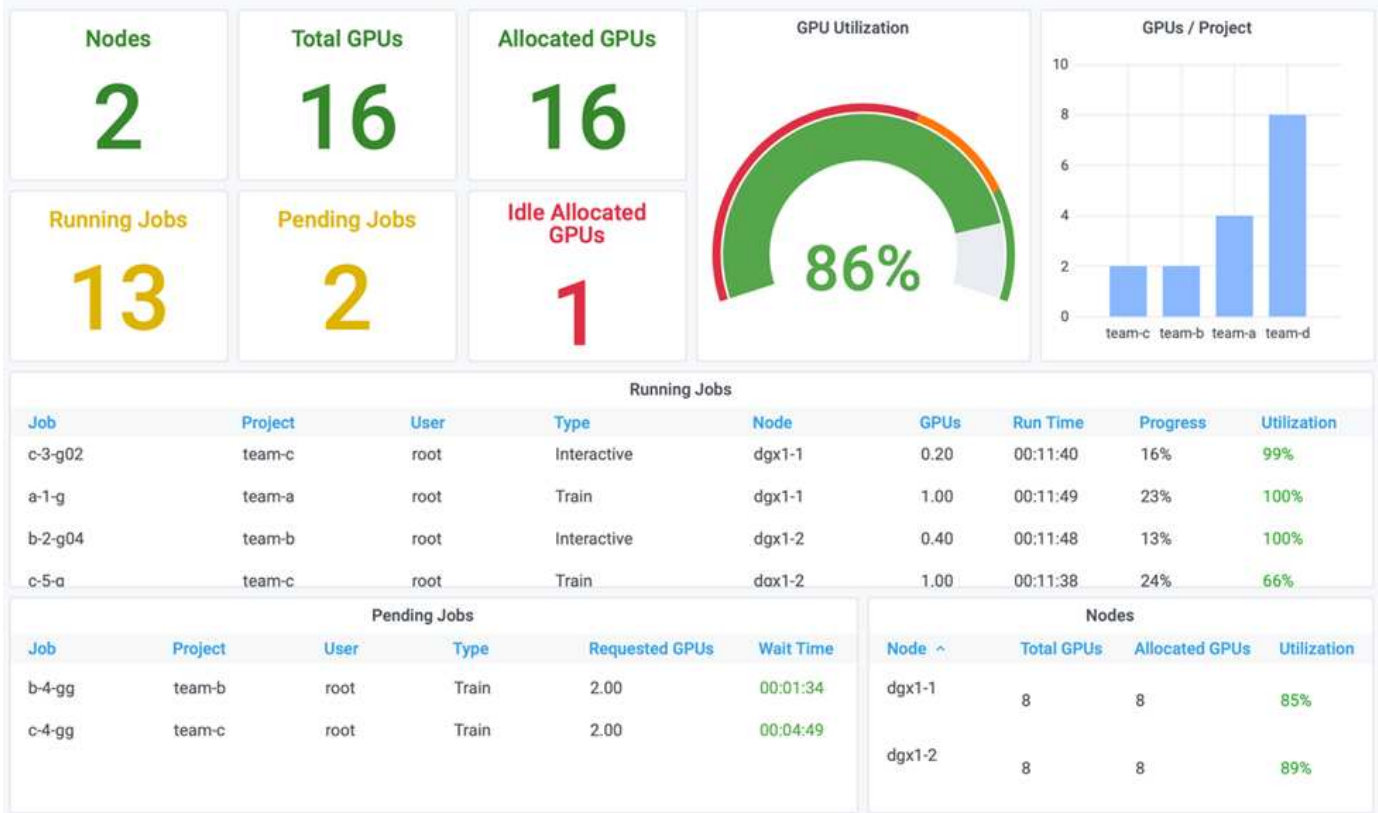
- 集群应已满：使用了 16/16 个 GPU。
- 集群利用率高。
- 由于分配百分比的影响，比 GPU 的实验更多。
- team-d 并未使用所有配额；因此，team-b 和 team-c 可以在其实验中使用额外的 GPU，从而加快创新速度。

基本资源分配公平

在本节中，我们会显示，当 team-d 请求更多 GPU（它们低于其配额）时，系统会暂停 team-b 和 team-c 的工作负载，并以公平的方式将其移至待定状态。

有关提交作业，使用的容器映像以及执行的命令序列等详细信息，请参见一节 ["第 4.9 节的测试详细信息"](#)。

下图显示了由于自动负载平衡和预先计划而产生的集群利用率，每个组分配的 GPU 以及待处理作业。我们可以观察到，当所有团队工作负载请求的 GPU 总数超过集群中可用的 GPU 总数时，Run：AI 的内部公平算法会分别为 team-b 和 team-c 暂停一个作业，因为它们已达到项目配额。这样可以提供整体较高的集群利用率，而数据科学团队仍在管理员设置的资源限制下工作。



此测试场景的结果显示以下内容：

- \* 自动负载均衡。\* 系统会自动平衡 GPU 的配额，使每个团队现在都在使用其配额。暂停的工作负载属于超过其配额的团队。
- \* 公平共享暂停。\* 系统会选择停止超过配额的一个组的工作负载，然后停止另一个组的工作负载。Run：AI 具有内部公平算法。

### 配额过度公平

在本节中，我们将扩展多个团队提交工作负载并超过其配额的情形。通过这种方式，我们将展示 Run：AI 的公平性算法如何根据预设备额比率分配集群资源。

此测试场景的目标：

- 显示多个团队请求超过其配额的 GPU 时的排队机制。
- 显示系统如何根据配额之间的比率在超过配额的多个组之间分配公平的集群份额，以便具有较大配额的组获得较大的备用容量份额。

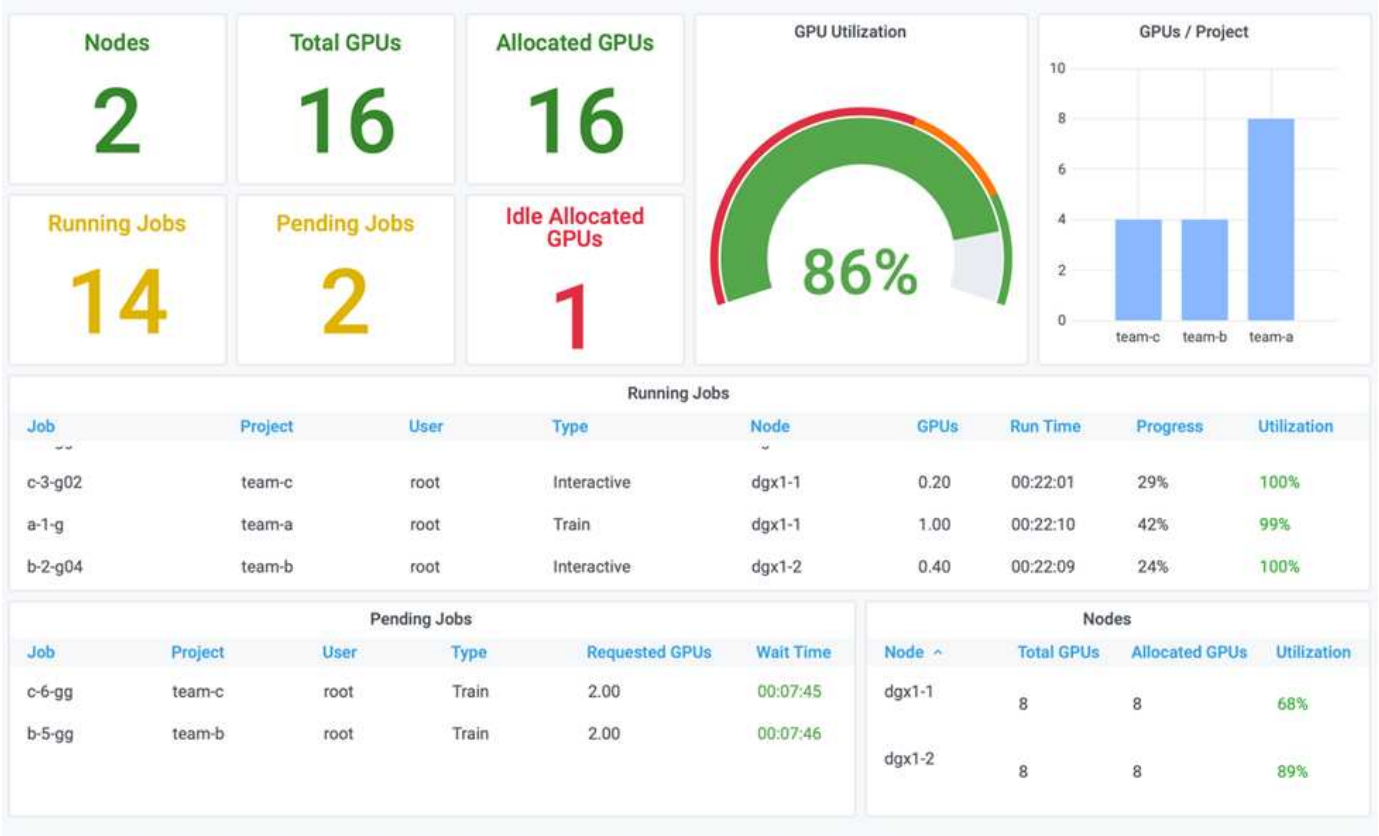
结束时 **"基本资源分配公平"**，有两个工作负载排队：一个用于 team-b，一个用于 team-c。在本节中，我们将对其他工作负载进行排队。

有关提交作业，使用的容器映像以及执行的命令序列等详细信息，请参见 **"第 4.10 节的测试详细信息"**。

根据部分提交所有作业时 **"第 4.10 节的测试详细信息"**，则系统信息板会显示此信息 team-a，team-b，和 team-c 所有处理器都具有比其预设备额更多的 GPU。team-a 所占用的 GPU 比其预设的软配额(四个)多四个、而 team-b 和 team-c 每个 CPU 占用的 GPU 比其软配额多两个(两个)。分配的过度配额 GPU 的比率等于其预设配额的比率。这是因为系统使用预设备额作为优先级的参考，并在多个团队请求更多 GPU，超过其配额时相应



地进行配置。当企业数据科学团队积极参与 AI 模型的开发和生产时，这种自动负载平衡可以实现公平和优先级划分。



此测试场景的结果如下：

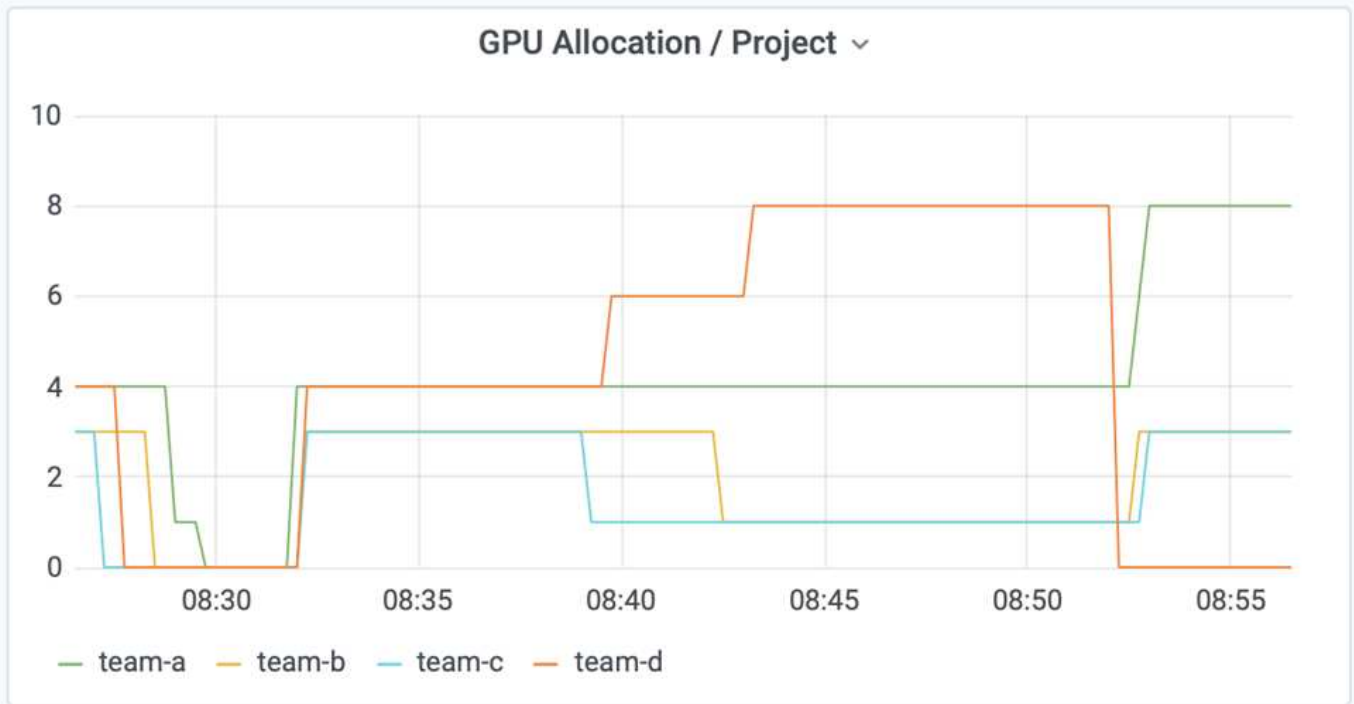
- 系统开始将其他团队的工作负载取消队列。
- 除队顺序根据公平算法来确定，这样 team-b 和 team-c 会获得相同数量的超配额 GPU（因为它们具有类似的配额），而 team-A 获得的 GPU 数量是原来的两倍，因为他们的配额是 team-b 和 team-c 的两倍。
- 所有分配都将自动完成。

因此，系统应在以下状态下保持稳定：

项目	已分配 GPU	comment
团队 A	8/4.	超过配额四个 GPU 。空队列。
团队 b	4/2	超过配额的两个 GPU 。一个工作负载已排队。
团队 c	4/2	超过配额的两个 GPU 。一个工作负载已排队。
团队	0/8	根本不使用 GPU ，没有已排队的工作负载。

下图显示了各个部分的 Run：AI Analytics 信息板中每个项目在一段时间内的 GPU 分配情况 "通过过度配额 GPU 分配实现高集群利用率"，"基本资源分配公平"，和 "配额过度公平"。图中的每一行表示在任何时间为给定数据科学团队配置的 GPU 数量。我们可以看到，系统会根据提交的工作负载动态分配 GPU。这样，当集群中

存在可用 GPU 时，团队可以超过配额，然后根据公平原则抢占作业，最后达到所有四个团队的稳定状态。



将数据保存到 **Trident** 配置的 **PersistentVolume**

NetApp Trident 是一个完全受支持的开源项目，旨在帮助您满足容器化应用程序的复杂持久性需求。您可以将数据读写到 Trident 配置的 Kubernetes PersistentVolume (PV) 中，并通过 NetApp ONTAP 数据管理软件提供数据分层，加密，NetApp Snapshot 技术，合规性和高性能优势。

重复使用现有命名空间中的 **PVC**

对于规模较大的 AI 项目，不同容器向同一个 Kubernetes PV 读取和写入数据可能会更高效。要重复使用 Kubernetes 永久性卷声明 (PVC)，用户必须已创建 PVC。请参见 ["NetApp Trident 文档"](#) 有关创建 PVC 的详细信息。以下是重复使用现有 PVC 的示例：

```
$ runai submit pvc-test -p team-a --pvc test:/tmp/pvc1mount -i gcr.io/run-ai-demo/quickstart -g 1
```

运行以下命令查看项目 team-A 的作业 pvc 测试 的状态：

```
$ runai get pvc-test -p team-a
```

您应看到 PV /tmp/pvc1mount 挂载到 team-A job vc-test。这样，多个容器就可以从同一个卷读取数据，这在开发或生产环境中存在多个竞争模式时非常有用。数据科学家可以构建一系列模型，然后通过多数投票或其他技术将预测结果结合起来。

使用以下命令访问容器 Shell：

```
$ runai bash pvc-test -p team-a
```

然后，您可以检查已挂载的卷并访问容器中的数据。

这种重复使用 PVC 的功能可与 NetApp FlexVol 卷和 NetApp ONTAP FlexGroup 卷配合使用，从而使数据工程师可以使用更灵活，更强大的数据管理选项来利用由 NetApp 提供支持的数据网络结构。

## 结论

NetApp 和 Run : AI 已在本技术报告中展开合作，展示了 NetApp ONTAP AI 解决方案与 Run : AI 平台在简化 AI 工作负载流程方面的独特功能。上述步骤提供了一个参考架构，用于简化深度学习的数据管道和工作负载流程。建议希望实施这些解决方案的客户联系 NetApp 和 Run : AI 了解更多信息。

## 第 4.8 节的测试详细信息

此部分包含此部分的测试详细信息 ["通过过度配额 GPU 分配实现高集群利用率"](#)。

按以下顺序提交作业：

项目	图像	GPU 数量	总计	comment
团队 A	Jupyter	1.	1 , 4	—
团队 A	NetApp	1.	2 , 4	—
团队 A	运行: AI	2.	4.4	正在使用其所有配额
团队 b	运行: AI	0.6	0.6/2	GPU 百分比
团队 b	运行: AI	0.4	1.2	GPU 百分比
团队 b	NetApp	1.	2/2	—
团队 b	NetApp	2.	4/2	两个超过配额
团队 c	运行: AI	0.5	0.5/2	GPU 百分比
团队 c	运行: AI	0 , 3.	0.8/2	GPU 百分比
团队 c	运行: AI	0.2	1.2	GPU 百分比
团队 c	NetApp	2.	3/2	一个超过配额
团队 c	NetApp	1.	4/2	两个超过配额
团队	NetApp	4.	4 , 8.	正在使用配额的一半

命令结构：

```
$ runai submit <job-name> -p <project-name> -g <#GPUs> -i <image-name>
```

测试中使用的实际命令顺序：

```

$ runai submit a-1-1-jupyter -i jupyter/base-notebook -g 1 \
  --interactive --service-type=ingress --port 8888 \
  --args="--NotebookApp.base_url=team-a-test-ingress" --command=start
-notebook.sh -p team-a
$ runai submit a-1-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-a
$ runai submit a-2-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a
$ runai submit b-1-g06 -i gcr.io/run-ai-demo/quickstart -g 0.6
--interactive -p team-b
$ runai submit b-2-g04 -i gcr.io/run-ai-demo/quickstart -g 0.4
--interactive -p team-b
$ runai submit b-3-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-b
$ runai submit b-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-b
$ runai submit c-1-g05 -i gcr.io/run-ai-demo/quickstart -g 0.5
--interactive -p team-c
$ runai submit c-2-g03 -i gcr.io/run-ai-demo/quickstart -g 0.3
--interactive -p team-c
$ runai submit c-3-g02 -i gcr.io/run-ai-demo/quickstart -g 0.2
--interactive -p team-c
$ runai submit c-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-c
$ runai submit c-5-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-c
$ runai submit d-1-gggg -i gcr.io/run-ai-demo/quickstart -g 4 -p team-d

```

此时，您应处于以下状态：

项目	已分配 <b>GPU</b>	已排队的工作负载
团队 A	4/4 （软配额 / 实际分配）	无
团队 b	4/2	无
团队 c	4/2	无
团队	4 ， 8.	无

请参见一节 ["通过过度分配 GPU 实现高集群利用率"](#) 用于讨论继续测试场景。

## 第 4.9 节的测试详细信息

此部分包含此部分的测试详细信息 ["基本资源分配公平"](#)。

按以下顺序提交作业：

项目	GPU 数量	总计	comment
团队	2.	6/8	team-b/c 工作负载暂停并移至 pending。
团队	2.	8/8.	其他团队（b/c）工作负载暂停并移至 pending。



请参见以下已执行的命令序列：

```
$ runai submit d-2-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-d$
runai submit d-3-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-d
```

此时，您应处于以下状态：

项目	已分配 GPU	已排队的工作负载
团队 A	4.4	无
团队 b	2/2	无
团队 c	2/2	无
团队	8/8.	无

请参见一节 ["基本资源分配公平"](#) 有关继续测试场景的讨论。

第 4.10 节的测试详细信息

此部分包含此部分的测试详细信息 ["配额过度公平"](#)。

按以下顺序提交 team-A ， team-b 和 team-c 的作业：

项目	GPU 数量	总计	comment
团队 A	2.	4.4	1 个工作负载已排队
团队 A	2.	4.4	已排队 2 个工作负载
团队 b	2.	2/2	已排队 2 个工作负载
团队 c	2.	2/2	已排队 2 个工作负载

请参见以下已执行的命令序列：

```
$ runai submit a-3-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a$
runai submit a-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a$ runai
submit b-5-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-b$ runai
submit c-6-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-c
```

此时，您应处于以下状态：

项目	已分配 GPU	已排队的工作负载
团队 A	4.4	两个工作负载要求每个 GPU 两个
团队 b	2/2	两个工作负载，每个工作负载需要两个 GPU

项目	已分配 GPU	已排队的工作负载
团队 c	2/2	两个工作负载，每个工作负载需要两个 GPU
团队	8/8.	无

接下来，删除 team-d 的所有工作负载：

```
$ runai delete -p team-d d-1-gggg d-2-gg d-3-gg
```

请参见一节 ["配额过度公平"](#)，用于讨论后续测试场景。

## 从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请参见以下资源：

- NVIDIA DGX 系统
  - NVIDIA DGX-1 系统<https://www.nvidia.com/en-us/data-center/dgx-1/>
  - NVIDIA V100 Tensor 核心 GPU<https://www.nvidia.com/en-us/data-center/tesla-v100/>
  - NVIDIA NGC<https://www.nvidia.com/en-us/gpu-cloud/>
- 运行： AI 容器编排解决方案
  - Run： AI 产品简介<https://docs.run.ai/home/components/>
  - 运行： AI 安装文档<https://docs.run.ai/Administrator/Cluster-Setup/Installing-Run-AI-on-an-on-premise-Kubernetes-Cluster/>  
<https://docs.run.ai/Administrator/Researcher-Setup/Installing-the-Run-AI-Command-Line-Interface/>
  - 在 Run： AI 命令行界面中提交作业<https://docs.run.ai/Researcher/Walkthroughs/Walkthrough-Launch-Unattended-Training-Workloads-/>  
<https://docs.run.ai/Researcher/Walkthroughs/Walkthrough-Start-and-Use-Interactive-Build-Workloads-/>
  - 在 Run： AI 命令行界面中分配 GPU 分数<https://docs.run.ai/Researcher/Walkthroughs/Walkthrough-Using-GPU-Fractions/>
- NetApp AI 控制平台
  - 技术报告<https://www.netapp.com/us/media/tr-4798.pdf>
  - 简短演示[https://youtu.be/gfr\\_sO27Rvo](https://youtu.be/gfr_sO27Rvo)
  - GitHub 存储库[https://github.com/NetApp/kubeflow\\_jupyter\\_pipeline](https://github.com/NetApp/kubeflow_jupyter_pipeline)
- NetApp AFF 系统
  - NetApp AFF A 系列产品规格<https://www.netapp.com/us/media/ds-3582.pdf>
  - 适用于全闪存 FAS 的 NetApp 闪存优势<https://www.netapp.com/us/media/ds-3733.pdf>
  - ONTAP 9 信息库<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>
  - NetApp ONTAP FlexGroup Volumes 技术报告<https://www.netapp.com/us/media/tr-4557.pdf>

- NetApp ONTAP AI

- 采用 DGX-1 的 ONTAP AI 和 Cisco 网络设计指南<https://www.netapp.com/us/media/nva-1121-design.pdf>
- 《采用 DGX-1 的 ONTAP AI 和 Cisco 网络部署指南》<https://www.netapp.com/us/media/nva-1121-deploy.pdf>
- 采用 DGX-1 和 Mellanox 网络设计指南的 ONTAP AI<http://www.netapp.com/us/media/nva-1138-design.pdf>
- 采用 DGX-2 的 ONTAP AI 设计指南<https://www.netapp.com/us/media/nva-1135-design.pdf>

## TR-4799-design: 适用于自主驱动工作负载的NetApp ONTAP AI参考架构

NetApp公司David Arnette和Sung-Han Lin

NVIDIA DGX系统系列是全球首款专为企业AI打造的集成人工智能(AI)平台。NetApp AFF 存储系统可提供极致性能和行业领先的混合云数据管理功能。NetApp和NVIDIA合作创建了NetApp ONTAP AI参考架构、为客户提供了一个统包解决方案、可为AI和机器学习(ML)工作负载提供企业级性能、可靠性和支持。

"TR-4799-design: 适用于自主驱动工作负载的NetApp ONTAP AI参考架构"

## TR-4811: 《适用于医疗保健的NetApp ONTAP AI参考架构: 诊断映像—解决方案 设计》

Rick Huang、Sung-Han Lin、Sathish Thyagarajan、NetApp JAcci Cenci、NVIDIA

此参考架构可为使用NVIDIA DGX-2系统和NetApp AFF 存储构建人工智能(AI)基础架构以满足医疗保健用例的客户提供指导。其中包括有关在为医学诊断成像、经验证的测试案例和结果开发深度学习(DL)模型时所用的高级工作流的信息。它还包括针对客户部署的规模估算建议。

"TR-4811: 《适用于医疗保健的NetApp ONTAP AI参考架构: 诊断映像—解决方案 设计》"

## TR-4807: 适用于金融服务工作负载的NetApp ONTAP AI参考架构—解决方案 设计

Karthikeyan Nagalingam、Sung-Han Lin、NetApp JAcci Cenci、NVIDIA

此参考架构为使用NVIDIA DGX-1系统和NetApp AFF 存储构建人工智能基础架构以满足金融部门使用情形的客户提供了准则。其中包括有关在为金融服务测试案例和结果开发深度学习模型时所用的高级工作流的信息。它还包括针对客户部署的规模估算建议。

"TR-4807: 适用于金融服务工作负载的NetApp ONTAP AI参考架构—解决方案 设计"

# 生成性AI和NetApp价值

对生成性人工智能(AI)的需求正在推动各行各业的变革、从而增强了业务创造力和产品创新。

作者：Sathish Thyagarajan、NetApp

## 摘要

许多企业都在使用生成型AI来构建新的产品功能、提高工程效率、并对AI驱动的应用程序进行原型设计、以提供更好的结果和消费者体验。生成型AI (例如生成型预训练变形本(GPT))使用神经网络创建新内容、包括文本、音频和视频。鉴于大型语言模型(LLM)所涉及的极大规模数据集、构建强大的AI基础架构至关重要、该基础架构应利用内部部署、混合云和多云部署选项的极具吸引力的数据存储功能、并降低与数据移动性相关的风险、在企业设计AI解决方案之前、先对数据进行保护和监管。本白皮书将介绍这些注意事项以及相应的NetApp@AI功能, 这些功能可以在人工智能数据管道之间实现无缝数据管理和数据移动, 以用于训练、再培训、微调 and 推理生成的人工智能模型。

## 内容提要

最近、在2022年11月推出GPT-3的衍生产品ChatGPT之后、用于根据用户提示生成文本、代码、图像甚至治疗性蛋白质的新AI工具获得了显著声誉。这表示用户可以使用自然语言提出请求、AI将使用经过训练的现有数据算法解释和生成文本、例如反映用户请求的新闻文章或产品说明、或者生成代码、音乐、语音、视觉效果和3D资产。因此、稳定扩散、幻想、快速工程和价值协调等短语在AI系统设计中迅速涌现。这些自我监督或半监督机器学习(ML)模式正在通过云服务提供商和其他AI公司供应商以预先训练的基础模式(FM)的形式广泛提供、这些模式正被各行各业的各种企业机构采用、用于执行各种下游NLP (自然语言处理)任务。正如McKinsey等研究分析公司所说:“生成性AI对生产率的影响可以为全球经济增加数以亿计的价值。”虽然企业正在将AI重新视为人类的思想合作伙伴、而FMS正在同时扩展到企业和机构可以利用生成型AI实现的目标、但管理海量数据的机会仍将继续增长。本文档介绍了有关生成AI的介绍信息、以及与可为NetApp客户带来价值的NetApp功能相关的设计概念、包括内部环境以及混合或多云环境。

那么、客户在AI环境中使用NetApp有何意义? NetApp可帮助企业应对数据和云快速增长、多云管理以及采用AI等下一代技术所带来的复杂性。NetApp已将各种功能整合到智能数据管理软件和存储基础架构中、这些功能与针对AI工作负载优化的高性能实现了完美平衡。像LMs这样的生成性AI解决方案需要多次将其源数据集从存储读取并处理到内存中、以增强智能。NetApp在跨边缘到核心再到云生态系统的数据移动性、数据监管和数据安全技术方面一直处于领先地位、为企业客户构建大规模AI解决方案提供服务。NetApp与强大的合作伙伴网络一直在帮助首席数据官、AI工程师、企业架构师和数据科学家设计自由流动的数据管道、用于数据准备、数据保护、以及AI模型训练和推理的战略数据管理职责、优化AI/ML生命周期的性能和可扩展性。NetApp数据技术和功能, 例如用于深度学习数据管道的NetApp@ONTAP AI®、用于在存储端点之间无缝高效地传输数据的NetApp@SnapMirror® NetApp@FlexCache®可在数据流从批量转换到实时时进行实时渲染, 并且数据工程会及时进行, 从而为部署实时生成AI模型带来价值。随着各种类型的企业纷纷采用新的AI工具、他们面临着从边缘到数据中心再到云端的数据挑战、这些挑战需要可扩展、负责任且可解释的AI解决方案。作为混合云和多云数据管理领域的权威企业、NetApp致力于构建合作伙伴网络和联合解决方案、帮助构建数据管道和数据湖的方方面面、以进行生成性AI模型训练(训练前)、微调、基于上下文的推理以及对LLM的模型衰退监控。

## 什么是生成型AI?









生成性AI正在改变我们创建内容、生成新设计概念以及探索新组成的方式。它展示了生成式抗压网络(GAN)、多种形式的自动编码器(VAE)和生成式预训练变形器(GPT)等神经网络框架、这些框架可以生成文本、代码、图像、音频、视频、和合成数据。OpenAI的Chat-GPT、Google的Bard、hubling face's bollama和Meta的llama等基于变压器的模型已成为支持大型语言模型许多进步的基础技术。同样、OpenAI的Dall-E、Meta的CM3leon和Google的Imagen也是文本到图像传播模型的示例、这些模型为客户提供前所未有的光刻度、让客户可以从头开始创建新的复杂图像、或者通过数据集扩充和文本到图像合成来编辑现有图像、从而生成高质量的上下文感知

图像。数字艺术家开始将Nerf (神经光场)等渲染技术与生成性AI相结合、将静态2D图像转换为沉浸式3D场景。一般来说、LLM的大致特征是四个参数：(1)模型大小(通常以数十亿个参数为单位)；(2)训练数据集大小；(3)训练成本；(4)训练后的模型性能。此外、LMs还主要分为三种变压器架构。(i)仅编码器型号。例如BERT (Google、2018年)；(ii)编码器-解码器、例如BART (Meta、2020年)和(iii)仅解码器型号。例如：llama (Meta、2023)、Palm/E (Google、2023)。根据业务要求、无论公司选择哪种架构、训练数据集中的模型参数数量(N)和令牌数量(D)通常都会确定训练(训练前)或微调LLM的基线成本。

企业用例和下游NLL任务

各行各业的企业都在发掘越来越多的AI潜力、从现有数据中提取并产生新形式的价值、用于业务运营、销售、营销和法律服务。根据IDC (International Data Corporation)在全球生成型AI用例和投资方面的市场情报、软件开发和产品设计方面的知识管理受到的影响最大、其次是为营销创建案例以及为开发人员生成代码。在医疗保健领域、临床研究组织正在医学领域开辟新天地。ProteinBERT等经过预先训练的模型采用基因本体(GGO)注释来快速设计医疗药物的蛋白质结构、这是药物发现、生物信息学和分子生物学方面的一个重要里程碑。生物技术公司已开始人类试验AI发现的生成性药物、其目的是治疗肺部成肿(IPF)等疾病、肺部成肿(IPF)是一种导致肺组织不可逆划痕的肺病。

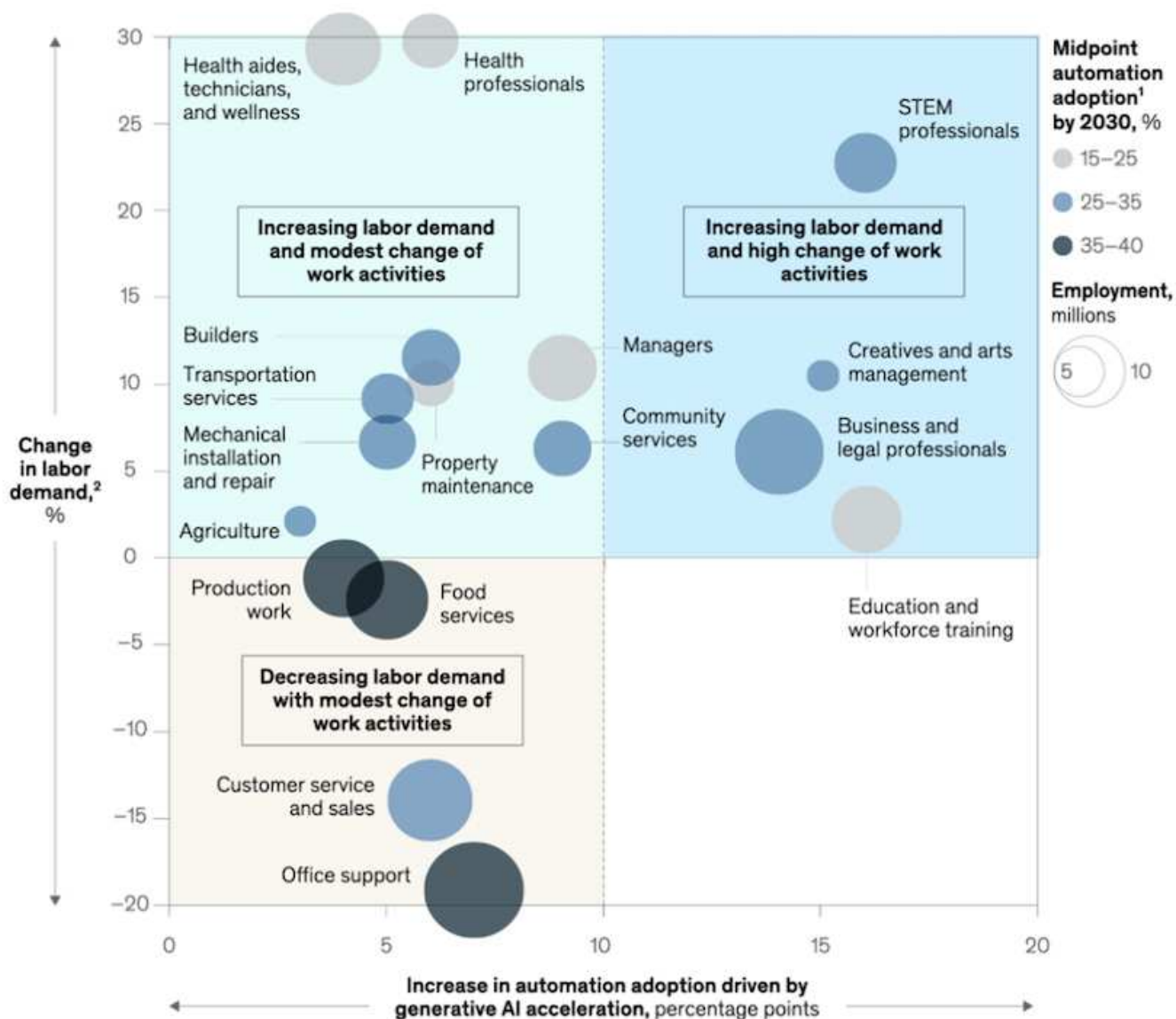
图1：推动生成性AI的用例

 <b>Chatbots</b>	 <b>Drug discovery</b>
 <b>Text generation</b>	 <b>Genome model expression</b>
 <b>Image generation</b>	 <b>Classification</b>
 <b>Code generation</b>	 <b>Speech-to-Text</b>

在生成型AI的推动下、自动化采用率的提高也在改变许多职业工作活动的供求。如McKinsey所述、美国劳动力市场(下图)经历了快速转型、只有在考虑到AI的影响后、这种转型才可能持续下去。

来源：McKinsey & Company

## Estimated labor demand change and generative AI automation acceleration by occupation, US, 2022–30



### 存储在生成AI中的作用

LLM在很大程度上依赖于深度学习、GPU和计算。但是、当GPU缓冲区填满时、需要将数据快速写入存储。虽然某些AI模型的大小足以在内存中执行、但LLM需要高IOPS和高吞吐量存储才能快速访问大型数据集、尤其是在涉及数十亿个令牌或数百万个图像的情况下。对于LLM的典型GPU内存需求、使用10亿个参数训练模型所需的内存最高可达80 GB @32位全精度。在这种情况下、Meta的Llama 2 (一个规模从70亿到700亿参数的LLM系列)可能需要70x80、大约5600 GB或5.6 TB GPU RAM。此外、所需的内存量与要生成的最大令牌数成正比。例如、如果要生成最多512个令牌(约380个字)的输出、则需要 "512 MB"。这可能看似无关紧要、但是、如果您要运行较大的批次、它就会开始累加。因此、组织在内存中进行训练或微调的成本非常高、从而使存储成为生成性AI的基石。

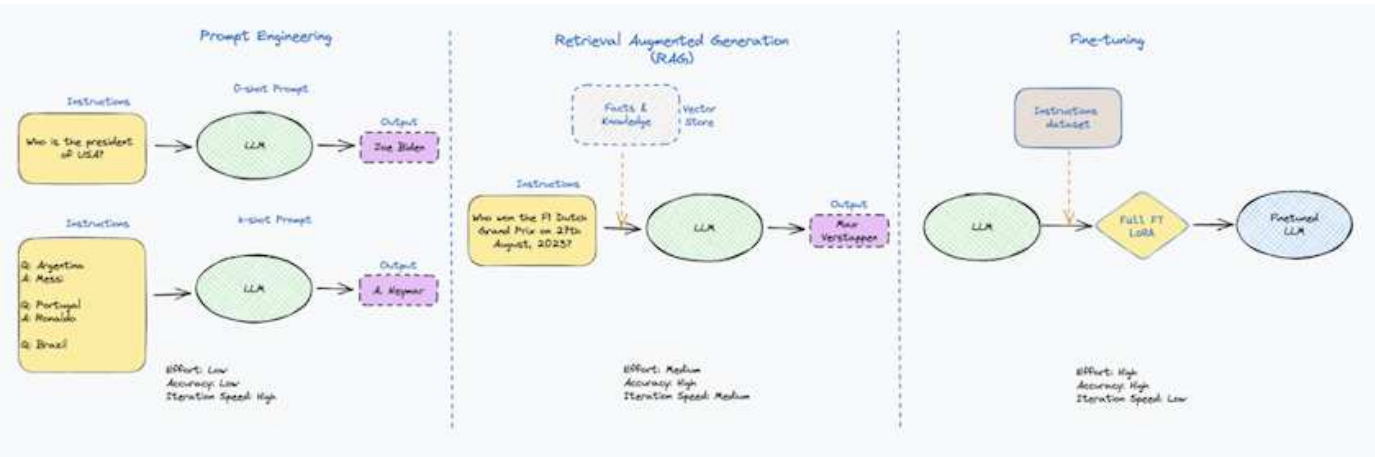
### 三种主要的LLMs方法

对于大多数企业而言、根据当前趋势、部署LLM的方法可以精简为3种基本方案。如最近的中所述 "《哈佛商业评论》" 文章：(1)从头开始培训(预培训) LLM—成本高昂、需要专业的AI/ML技能；(2)利用企业数据微调基础模型—复杂但可行；(3)使用检索增强生成(RAG)查询包含公司数据的文档存储库、API和矢量数据库。其中每一种方法都在其实施过程中的工作量、迭代速度、成本效益和模型准确性之间进行权衡、以解决不同类型的问题(下



图)。

图3：问题类型



基础模型

基础模型(FM)也称为基础模型(Base Model)、它是一种大型AI模型(LLM)、使用大规模自我监督、针对大量下游NLP"任务进行训练。由于训练数据不是由人类标记的、因此模型会出现、而不是显式编码。这意味着模型可以生成自己的故事或叙述、而无需明确编程。因此、FM的一个重要特征是同质化、这意味着在许多领域中使用相同的方法。但是、借助个性化和微调技术、如今出现的产品中集成的FMS不仅能够很好地生成文本、文本到图像和文本到代码、还可以解释特定域的任务或调试代码。例如、FMS (如OpenAI的Codex"或Meta的Code Llama)可以根据编程任务的自然语言描述以多种编程语言生成代码。这些模型精通十几种编程语言、包括Python、C#、JavaScript、Perl、Ruby、和SQL。他们了解用户的意图、并生成特定的代码来完成对软件开发、代码优化和编程任务自动化有用的预期任务。

微调、特定域和重新培训

在数据准备和数据预处理之后、LLM部署的一个常见做法是、选择一个经过预先训练的模型、该模型已在庞大且多样化的数据集中进行过训练。在微调环境中、这可以是一个开源大型语言模型、例如 "元数据的Llama 2." 经过700亿个参数和2万亿个令牌的培训。选择预先训练的模型后、下一步是根据域特定的数据对其进行微调。这包括调整模型的参数并对其进行新数据训练、以适应特定的领域和任务。例如、作为一家专有LLM公司的布隆伯格GPT、就为金融行业提供的各种金融数据进行了培训。为特定任务设计和训练的域特定模型在其范围内的准确性和性能通常较高、但在其他任务或域之间的可转移性较低。当业务环境和数据在一段时间内发生变化时、与测试期间的性能相比、FM的预测准确性可能开始下降。此时、重新培训或微调模型变得至关重要。传统人工智能/机器学习中的模型再训练是指使用新数据更新已部署的机器学习模型、通常执行此操作是为了消除出现的两种类型的移动。(1)概念漂移—当输入变量和目标变量之间的链接随时间发生变化时、由于我们要预测的内容的问题描述发生变化、因此模型可能会产生不准确的预测。(2)数据漂移—当输入数据的特征发生变化时、例如客户习惯或行为随时间推移而发生变化、因此模型无法响应此类变化。以类似的方式、重新培训适用场景FMS/LLM、但是成本可能会很高(以百万美元为单位)、因此大多数企业可能不会考虑这一点。它正在积极研究、仍在LLMOps领域出现。因此、当模型在微调FMS中发生退换时、企业可能会选择使用较新的数据集再次进行微调(成本低得多)。从成本角度来看、下面列出了Azazy-OpenAI Services的模型价格表示例。对于每个任务类别、客户可以微调和评估特定数据集的模型。

来源：Microsoft Azure



Model	Per 1000 token
Text-Ada	\$0.0001
GPT-3.5 Turbo	\$0.003
GPT-4	\$0.06
Text-Davinci	\$0.02
Model	Per 100 images
Dall-E	\$2

提示工程和推理

提示工程是指在不更新模型权重的情况下、如何与LLM进行通信以执行所需任务的有效方法。与AI模型训练和微调一样重要的是NLG应用程序、推理也同样重要、因为经过训练的模型会响应用户提示。推理的系统要求通常更多地涉及AI存储系统的读取性能、该存储系统会将数据从LLM馈送到GPU、因为它需要能够应用数十亿个已存储的模型参数来生成最佳响应。

LLMOps、Model Monitoring和Mittorstores

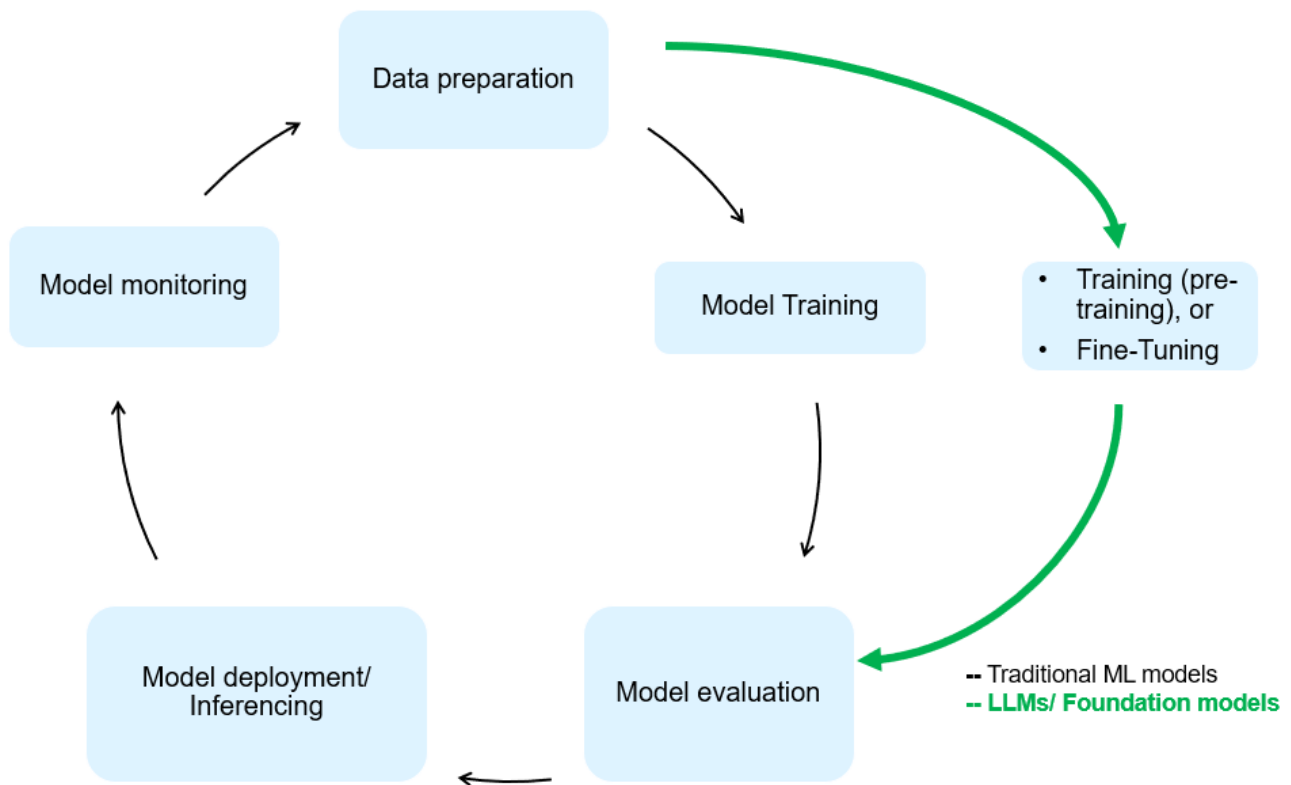
与传统机器学习操作(MLOps)一样、大型语言模型操作(LLMOps)也需要数据科学家和DevOps工程师协作、利用工具和最佳实践在生产环境中管理LLM。但是、LLM的工作流和技术堆栈可能会在某些方面有所不同。例如、使用LangChin字符串等框架构建的LLM管道将对外部嵌入端点(如矢量存储库或向量数据库)的多个LLM API调用组合在一起。将嵌入端点和矢量存储用于下游连接器(如向量数据库)代表了数据存储和访问方式的重大发展。与从零开始开发的传统ML模型不同、LMs通常依赖于转移学习、因为这些模型从FMS开始、FMS会根据新数据进行微调、以提高更具体领域的性能。因此、LLMOps提供风险管理和模型核降监测功能至关重要。

在生成型AI时代的风险与道德

"ChatGPT-它很流畅、但仍不实用。"-MIT Tech Review。垃圾输入-垃圾输出一直是计算领域的难题。与生成型AI的唯一区别在于、它擅长使垃圾变得高度可信、从而导致结果不准确。LLM倾向于根据自己的叙述来创造事实。因此、如果公司将生成型AI视为使用AI等效产品降低成本的绝佳机会、则需要高效检测深度假象、减少偏见并降低风险、以保持系统的诚信和道德。在设计负责任且可解释的生成型AI模型时、采用支持数据移动性、数据质量、数据监管和数据保护的强大AI基础架构的自由流动数据管道是一项杰出的功能。

客户场景和NetApp

图3：机器学习/大型语言模型 workflow



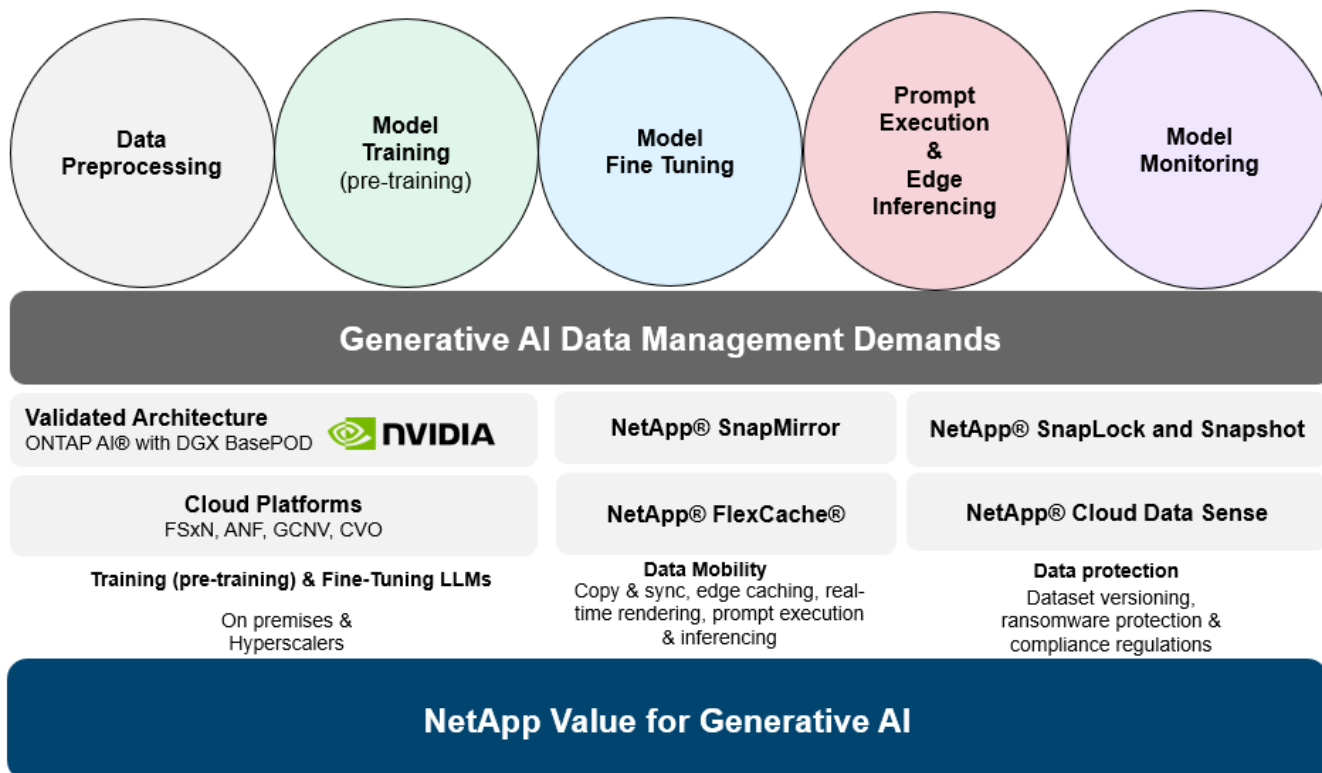
\*我们是在训练还是微调？\*是(a)从头开始训练LLM模型、微调预先训练的FM、还是使用RAG从基础模型以外的文档库中检索数据并增加提示、(b)无论是利用开源LMs (例如Llama 2)还是专有FMS (例如ChatGPT、Bard、AWS Brock)、都是企业的一项战略决策。每种方法都会在成本效益、数据惯性、操作、模型准确性和LLM管理之间进行折让。

作为一家公司、NetApp在其内部工作文化以及产品设计和工程工作方法中都采用AI。例如、NetApp的自主勒索软件保护功能是使用AI和机器学习构建的。它可以及早检测文件系统异常情况、帮助您在威胁影响操作之前识别这些威胁。其次、NetApp在销售和库存预测以及聊天机器人等业务运营中使用预测性AI、在呼叫中心产品支持服务、技术规格、保修、服务手册等方面为客户提供帮助。第三、NetApp通过为客户提供服务的产品和解决方案为AI数据管道和ML/LLM workflow带来客户价值构建预测性AI解决方案、例如需求预测、医学成像、情感分析、和生成性AI解决方案(如用于制造业图像异常检测的GANS)，以及银行和金融服务中的反洗钱和欺诈检测，均采用NetApp®ONTAP AI®、NetApp®SnapMirror®和NetApp®FlexCache®等NetApp产品和功能。

## NetApp功能

在聊天机器人、代码生成、图像生成或基因组模型表达等生成型AI应用程序中移动和管理数据可以跨越边缘、私有数据中心和混合多云生态系统。例如、通过ChatGPT等经过预先训练的模型的API公开的最终用户应用程序、帮助乘客将机票升级到商务舱的实时人工智能机器人无法自行完成此任务、因为乘客信息不会在互联网上公开。API要求从航空公司访问乘客的个人信息和机票信息、这些信息可能存在于混合云或多云生态系统中。类似的情形可能适用于科学家通过最终用户应用程序共享药物的一种药物和患者数据、该应用程序使用LLM在涉及一对多生物医学研究机构的药物发现过程中完成临床试验。传递给FMS或LLM的敏感数据可能包括：可识别身份信息、财务信息、运行状况信息、生物识别数据、位置数据、通信数据、在线行为和法律信息。在这种实时渲染、快速执行和边缘推理事件中、数据会通过开源或专有LLM模型从最终用户应用程序移动到存储端点、然后移动到内部或公共云平台上的数据中心。在所有这类场景中、数据移动性和数据保护对于涉及LLM的AI操作至关重要、因为这类操作依赖于大型训练数据集和此类数据的移动。

图4：生成型AI - LLM数据管道



NetApp的存储基础架构、数据和云服务产品组合由智能数据管理软件提供支持。

数据准备：LLM技术堆栈的第一个支柱与旧的传统ML堆栈基本没有任何不同。AI管道中的数据预处理对于在训练或微调之前对数据进行规范化和清理至关重要。此步骤包括用于以Amazon S3层形式或内部存储系统(例如文件存储或对象存储(例如NetApp StorageGRID)中的任何位置导入数据的连接器。

\*NetApp NetApp®ONTAP \*是NetApp在数据中心和云中的关键存储解决方案的基础技术。ONTAP包括各种数据管理和保护特性和功能、包括针对网络攻击的自动勒索软件保护、内置数据传输特性以及适用于各种架构的存储效率功能、从NAS、SAN、对象、LLM部署的软件定义的存储(SDS)情况。

用于深度学习模型训练的**NetApp®ONTAP AI®**。NetApp®ONTAP®支持使用基于RDMA的NFS的NVIDIA GPU Direct Storage™，适用于具有ONTAP存储集群和NVIDIA DGX计算节点的NetApp客户。它可以经济高效地将源数据集从存储读取并处理多次到内存中、以增强智能、从而使组织能够通过培训、微调和扩展对LLM的访问。

- NetApp®FlexCache®\*是一种远程缓存功能，它可以简化文件分发并仅缓存正在读取的数据。这对于LLM培训、再培训和微调非常有用、可以为具有实时渲染和LLM推理等业务需求的客户带来价值。
- NetApp®SnapMX\*是一种ONTAP功能，可在任意两个ONTAP系统之间复制卷快照。此功能可以以最佳方式将边缘数据传输到内部数据中心或云。如果客户希望在包含企业数据的RAG中开发生成性AI、则可以使用SnapMirror在内部云和超大型云之间安全高效地移动数据。它可以仅高效传输更改、节省带宽并加快复制速度、从而在FMS或LLM的训练、重新训练和微调操作期间提供基本的数据移动功能。

\*ONTAP®SnapLock为基于NetApp的存储系统提供了不可变的磁盘功能，用于数据集版本控制。微核架构旨在通过FPolicy™Zero Trust引擎保护客户数据。当攻击者以特别消耗资源的方式与LLM交互时、NetApp可通过抵御拒绝服务(DoS)攻击来确保客户数据可用。

- NetApp®云数据感知\*有助于识别、映射和分类企业数据集中的个人信息、制定策略、满足内部或云中的隐私要求、帮助改进安全防护并遵守法规。

\*由NetApp Data Sense提供支持的Cloud®BlueXP™\*分类。客户可以自动扫描、分析、分类和处理数据资产中的数据、检测安全风险、优化存储并加快云部署速度。它通过统一控制平台将存储和数据服务结合在一起、客户

可以使用GPU实例进行计算、并使用混合多云环境进行冷存储分层以及归档和备份。

NetApp文件-对象双重性\*。NetApp ONTAP支持对NFS和S3进行双协议访问。借助此解决方案、客户可以通过NetApp Cloud Volumes ONTAP的S3存储分段从Amazon AWS SageMaker笔记本电脑访问NFS数据。这为需要轻松访问异构数据源并能够共享NFS和S3数据的客户提供了灵活性。例如、在SageMaker上对FMS进行微调、例如可以访问文件对象分段的Meta的Llama 2文本生成模型。

- NetApp@Cloud Sync \*服务提供了一种简单安全的方法，可以将数据迁移到云端或内部环境中的任何目标。Cloud Sync可在内部或云存储、NAS和对象存储之间无缝传输和同步数据。

\*NetApp XCP\*是一款客户端软件，支持快速、可靠地将任何数据迁移到NetApp和NetApp数据迁移到NetApp。XCP还可以高效地将批量数据从Hadoop HDFS文件系统移动到ONTAP NFS、S3或StorageGRID中、而XCP文件分析可提供文件系统可见性。

- NetApp@DataOps Toolkit\*是一个Python库，数据科学家、开发运营人员和数据工程师可以利用它轻松地执行各种数据管理任务，例如近乎瞬时地配置、克隆或快照数据卷或JupyterLab工作空间，这些工作空间由高性能横向扩展NetApp存储提供支持。

**NetApp**的产品安全性。LLM可能会无意中在其响应中泄露机密数据、因此对于研究与利用LLM的AI应用程序相关的漏洞的ISO来说、这是一个顾虑。正如开放全球应用程序安全项目(Open Worldwide Application Security Project、Open Worldwide Application Security Project)所述、数据中毒、数据泄露、拒绝服务和在LLM中迅速注入等安全问题可能会因数据暴露给未经授权的访问服务攻击者而影响企业。数据存储要求应包括结构化、半结构化和非结构化数据的完整性检查和不可变更的快照。NetApp快照和SnapLock用于数据集版本控制。它提供了严格的基于角色的访问控制(Role-Based Access Control、RBAC)以及安全协议和行业标准加密、用于保护空闲和传输中的数据。Cloud Insights和Cloud Data Sense相结合、可帮助您预先确定威胁源并确定要还原的数据的优先级。

### **\*采用DGX BasePOD\*的ONTAP AI**

采用NVIDIA DGX BasePOD的NetApp@ONTAP@AI参考架构是一种适用于机器学习(ML)和人工智能(AI)工作负载的可扩展架构。在LLM的关键训练阶段、通常会定期将数据从数据存储复制到训练集群中。此阶段使用的服务器使用GPU来并行处理各种数据、从而产生巨大的数据需求。满足原始I/O带宽需求对于保持高GPU利用率至关重要。

### **\*NVIDIA AI企业版的ONTAP AI**

NVIDIA AI Enterprise是一款端到端云原生AI和数据分析软件套件、经过NVIDIA优化、认证和支持、可在采用NVIDIA认证系统的VMware vSphere上运行。此软件有助于在现代混合云环境中轻松快速地部署、管理和扩展AI工作负载。由NetApp和VMware提供支持的NVIDIA AI Enterprise通过一个简单熟悉的软件包提供企业级AI工作负载和数据管理。

### **1P云平台**

完全托管的云存储产品以Azure NetApp Files (ANF)的形式在Microsoft Azure上提供、以Amazon FSx for NetApp ONTAP (FSxN)的形式在AWS上提供、以Google Cloud NetApp Volumes (GNCV)的形式在Google上提供。1P是一款高性能托管文件系统、支持客户在公有云中运行高可用性AI工作负载、并提高数据安全性、以便使用AWS SageMaker、Azure-OpenAI Services和Google VertexAI等云原生ML平台微调LLM/FMS。

## **NetApp合作伙伴解决方案套件**

除了核心数据产品、技术和功能之外、NetApp还与强大的AI合作伙伴网络密切合作、为客户带来附加价值。

\*人工智能系统中的NVIDIA防护装置\*是确保以合乎道德和负责任的方式使用人工智能技术的保障措施。AI开发人

员可以选择定义基于LLM的应用程序在特定主题上的行为、并防止他们参与有关不需要的主题的讨论。护栏是一个开源工具包、能够无缝安全地将LLM连接到其他服务、从而构建可靠、安全的LLM对话系统。

\*Domino Data Lab\*提供多用途企业级工具，用于快速、安全、经济地构建生成型AI并将其产品化，无论您处于AI之旅的哪个阶段。借助Domino的企业MLOps平台、数据科学家可以使用首选工具及其所有数据、随时随地轻松训练和部署模型、并经济高效地管理风险-所有这些都从一个控制中心完成。

**Modzy for Edge AI。**NetApp®和Modzy携手合作，为任何类型的数据(包括图像、音频、文本和表格)提供大规模AI。Modzy是一个MLOps平台、用于部署、集成和运行AI模型、为数据科学家提供了模型监控、漂移检测和可解释性的功能、并集成了解决方案以实现无缝的LLM推定。

\*Run: AI\*和NetApp携手合作、展示NetApp ONTAP AI解决方案与Run: AI集群管理平台的独特功能、以简化AI工作负载的流程编排。它可以自动拆分和连接GPU资源、利用适用于Spark、Ray、dask和Rapids的内置集成框架将数据处理管道扩展到数百台计算机。

## 结论

只有在模型基于大量高质量数据进行训练后、生成型AI才能产生有效结果。虽然LMs已经取得了令人瞩目的里程碑式成就、但必须认识到其与数据移动性和数据质量相关的局限性、设计挑战和风险。LMs依赖于来自异构数据源的大型且不同的训练数据集。模型产生的不准确结果或有偏见的结果可能会使企业和消费者处于危险之中。这些风险可能与LLM因数据质量、数据安全性和数据移动性相关的数据管理挑战而面临的限制相对应。NetApp可帮助企业应对因数据快速增长、数据移动性、多云管理和采用AI而带来的复杂性。大规模AI基础架构和高效的数据管理对于定义生成型AI等AI应用程序的成功至关重要。关键在于、客户必须涵盖所有部署情形、同时不影响根据企业需求进行扩展的能力、同时保持成本效益、数据监管和合乎道德的AI实践控制权。NetApp一直致力于帮助客户简化和加快AI部署。

## TR-4785：使用NetApp E系列和BeeGFS进行AI部署

Nagalakshmi Raju、Daniel Landes、Nathan Swartz、NetApp公司Amind本纳尼

人工智能(AI)、机器学习(ML)和深度学习(DL)应用程序涉及大型数据集和高计算。要成功运行这些工作负载、您需要一个灵活的基础架构、使您能够无缝地横向扩展存储和计算节点。本报告介绍了在分布式模式下运行AI训练模型的步骤、该模式允许无缝横向扩展计算和存储节点。该报告还包括各种性能指标、用于显示将NetApp E系列存储与BeeGFS并行文件系统相结合的解决方案 如何为AI工作负载提供灵活、经济高效且简单的解决方案。

["TR-4785：使用NetApp E系列和BeeGFS进行AI部署"](#)

## NVA-1150-设计：《采用NetApp E系列系统的Quantum StorNext》设计指南

NetApp公司Ryan Rodine

本文档详细介绍了如何使用NetApp E系列存储系统设计StorNext并行文件系统解决方案。此解决方案 涵盖NetApp EF280全闪存阵列、NetApp EF300全闪存NVMe阵列、EF600全闪存NVMe阵列和NetApp E5760混合系统。它基于Frametest基准测试提供性能特征化、该工具广泛用于媒体和娱乐行业的测试。

["NVA-1150-设计：《采用NetApp E系列系统的Quantum StorNext》设计指南"](#)

# NVA-1150-Deploy: 《采用NetApp E系列系统的Quantum StorNext部署指南》

NetApp公司Ryan Rodine

本文档详细介绍了如何在NetApp E系列存储系统中部署StorNext并行文件系统解决方案。此解决方案 涵盖NetApp EF280全闪存阵列、NetApp EF300全闪存NVMe阵列、NetApp EF600全闪存NVMe阵列和NetApp E5760混合系统。它基于Frametest基准测试提供性能特征化、该工具广泛用于媒体和娱乐行业的测试。

"NVA-1150-Deploy: 《采用NetApp E系列系统的Quantum StorNext部署指南》 "



## 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。