



使用案例 NetApp Solutions

NetApp
April 12, 2024

目錄

使用案例	1
負責的AI和機密推斷資料-採用Protopia Image Transformation的NetApp AI	1
運用NetApp AI進行情緒分析	24
Azure的分散式訓練：點選率預測	41
TR-4896：Azure的分散式訓練：線道偵測-解決方案設計	63
TR-4841：混合雲AI作業系統、含資料快取	91
AI在Edge大會上推斷- NetApp採用Lenovo ThinkSystem -解決方案設計	112
WP-7328：使用NVIDIA JARVIS的NetApp對話AI	133
TR-4858：NetApp協調解決方案搭配Run:AI	152
TR-4799-Design：適用於ONTAP 自動駕駛工作負載的NetApp AI參考架構	170
TR-4811：適用於ONTAP 醫療的NetApp SAI參考架構：診斷影像-解決方案設計	170
TR-4807：ONTAP 適用於金融服務工作負載的NetApp AI參考架構-解決方案設計	170
推廣 AI 和 NetApp 價值	170
TR-4785：採用NetApp E系列和BeeGFS的AI部署	179
NVA-1150-Design：Quantum StorNext with NetApp E系列系統設計指南	179
NVA-1150-Deploy：Quantum StorNext with NetApp E系列系統部署指南	180

使用案例

負責的AI和機密推斷資料-採用Protopia Image Transformation的NetApp AI

TR-4928：負責的AI與機密資料推斷-採用Protopia Image與資料轉型的NetApp AI

Sathish Thyagarajan、Michael Oglesby、NetApp秉恩、Jennifer Cwagberg、Protopia

視覺判讀已成為溝通不可或缺的一部分、因為影像擷取和影像處理的出現。數位影像處理中的人工智慧（AI）帶來嶄新商機、例如癌症醫學領域和其他疾病辨識、研究環境危害的地理空間視覺分析、模式辨識、打擊犯罪的影片處理等。然而、這次機會也帶來非凡的責任。

企業組織將越多決策交給AI、就越能接受與資料隱私和安全性、以及法律、道德和法規問題有關的風險。負責的AI能讓公司和政府組織建立信任與治理、這對於大型企業的AI規模而言至關重要。本文件說明NetApp在三種不同案例中驗證的AI推斷解決方案、其方法是將NetApp資料管理技術與Protopia資料混淆軟體搭配使用、以將敏感資料私有化、並降低風險和道德疑慮。

消費者和企業實體每天都會透過各種數位裝置產生數百萬個影像。隨著資料與運算工作負載的大規模爆炸、企業紛紛改用雲端運算平台來實現擴充性與效率。同時、將影像資料中所含的敏感資訊傳輸至公有雲時、也會產生隱私顧慮。缺乏安全性與隱私權保證、成為部署映像處理AI系統的主要障礙。

此外、還有 "**銷毀權利**" 根據GDPR、個人有權要求組織清除所有個人資料。也有 "**隱私權法案**"，建立公平資訊實務守則。照片等數位影像可能構成GDPR規定的個人資料、而GDPR規定必須如何收集、處理及清除資料。若未遵守GDPR、可能會因違反法規而導致高額罰款、嚴重傷害組織。隱私權原則是實作負責AI的骨幹、可確保機器學習（ML）和深度學習（DL）模型預測的公平性、並降低違反隱私權或法規遵循的相關風險。

本文件說明已通過驗證的設計解決方案、可在三種不同的情境下、使用或不使用與維護隱私權及部署負責AI解決方案相關的影像模糊化：

- 情境1. Jupyter筆記型電腦內的隨需推斷。
- *案例2.*在Kubernetes上進行批次推斷。
- 情境3. NVIDIA Triton推斷伺服器。

針對此解決方案、我們使用專為研究不受限制的面偵測問題而設計的面向區域資料集（Fddb）、以及用於實作FaceBoxes的PyTorch機器學習架構。此資料集包含一組2845個不同解析度影像中5171個面的註釋。此外、本技術報告也提供一些解決方案領域、以及在適用本解決方案的情況下、從NetApp客戶和現場工程師收集到的相關使用案例。

目標對象

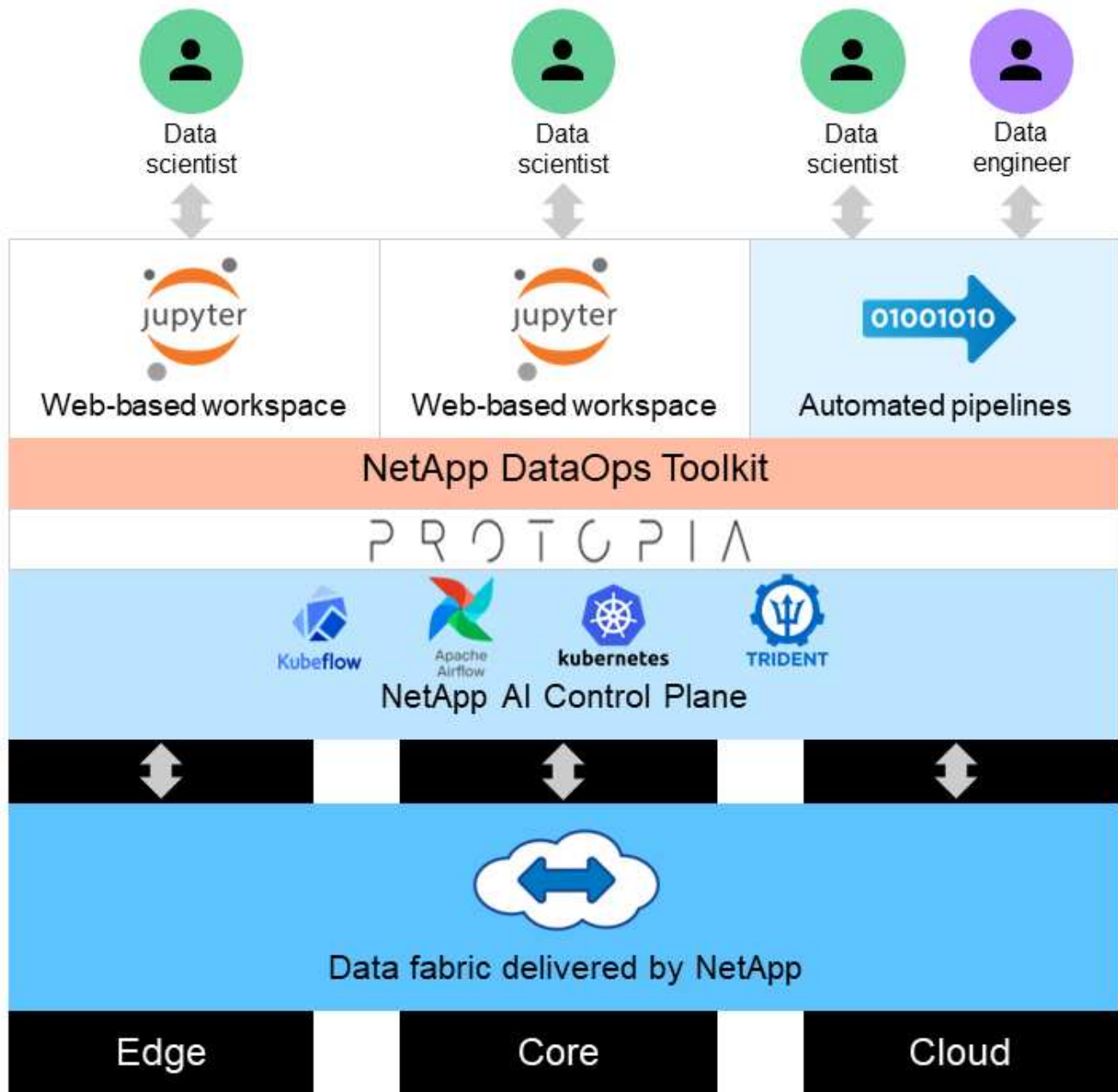
本技術報告適用於下列對象：

- 想要設計及部署負責AI、並解決公共空間中的面向影像處理相關資料保護與隱私權問題的企業領導者與企業架構設計師。
- 資料科學家、資料工程師、AI /機器學習（ML）研究人員、以及致力保護及維護隱私權的AI/ML系統開發人員。

- 企業架構設計師、為符合GDPR、CCPA或國防部隱私權法案（DoD）和政府組織等法規標準的AI/ML模型和應用程式設計資料混淆解決方案。
- 資料科學家和AI工程師正在尋求有效方法來部署深度學習（DL）和AI / ML / DL推斷模型、以保護敏感資訊。
- 負責部署及管理Edge推斷模型的Edge裝置管理員和Edge Server管理員。

解決方案架構

此解決方案的設計、是為了處理大型資料集上的即時和批次推斷AI工作負載、其處理能力是利用GPU與傳統CPU的處理能力。這項驗證可證明對於ML的隱私保護推論、以及尋求負責AI部署的組織所需的最佳資料管理。此解決方案提供適用於單一或多節點Kubernetes平台的架構、可用於邊緣與雲端運算、透過ONTAP 內部部署核心的NetApp AI、NetApp DataOps Toolkit、以及使用Jupyter Lab和CLI介面的Protopia混淆軟體進行互連。下圖顯示採用DataOps Toolkit和Protopia技術之NetApp技術的資料架構邏輯架構總覽。



Protopia混淆軟體可在NetApp DataOps Toolkit上順暢執行、並在離開儲存伺服器之前轉換資料。

解決方案領域

數位影像處理有許多優點、讓許多組織能夠充分利用視覺呈現的相關資料。此NetApp與Protopia解決方案提供獨特的AI推斷設計、可在ML/DL生命週期內保護及私有化AI/ML資料。它可讓客戶保留敏感資料的所有權、運用公有雲或混合雲部署模式來擴充規模和提高效率、減輕隱私權相關疑慮、並在邊緣部署AI推斷。

環境情報

產業在環境危害領域中、有許多方法可以利用地理空間分析。政府和公共工程部門可針對公共衛生和天氣狀況、提出可行的見解、以便在流行病或天然災難（例如野火）期間、更好地向大眾提供建議。例如、您可以在機場或醫院等公共空間中識別COVID-正向病患、而不會影響受影響個人的隱私、並提醒鄰近的相關機關和大眾採取必要的安全措施。

邊緣裝置穿戴式裝置

在軍事領域和戰場上、您可以使用邊緣的AI推斷裝置作為穿戴式裝置、來追蹤戰士的健康狀況、監控駕駛行為、並向主管機關警示接近軍事車輛的安全風險及相關風險、同時保留及保護戰士的隱私。軍事的未來將透過戰場物聯網（IoBT）和軍事物聯網（IoMT）來發展高科技、以提供穿戴式戰鬥裝備、協助戰士利用快速邊緣運算來識別敵人、並在戰鬥中表現更好。保護和保留從無人機和穿戴式設備等邊緣設備收集的視覺資料、對於讓駭客和敵人留在海灣是至關重要的。

非戰鬥人員清空作業

非戰鬥人員清空作業（NEO）由國防部執行、以協助疏散美國公民和國民、DOD文職人員、以及生命面臨適當安全避風港危險的指定人員（接待國（HN）和第三國國民（TCN）。已實施的管理控管措施主要是手動進行疏散者篩選程序。不過、使用高度自動化的AI/ML工具搭配AI / ML影片混淆技術、可能會改善疏散人員識別、疏散追蹤及威脅篩選的準確度、安全性及速度。

醫療與生物醫學研究

影像處理是用來診斷從電腦斷層掃描（CT）或磁共振造影（MRI）取得的3D影像進行手術規劃的不正常情況。HIPAA隱私權規則規範組織如何收集、處理及清除所有個人資訊和數位影像（例如照片）的資料。為了讓資料符合HIPAA安全港規定的可共享資格、必須移除全面相片影像和任何類似影像。用於從結構性CT/磁共振影像中隱藏個人的面向功能的自動辨識或頭骨刪除演算法等技術、已成為生物醫學研究機構資料共享程序的重要一環。

AI / ML分析的雲端移轉

企業客戶過去曾接受過內部部署的AI/ML模式訓練與部署。基於規模經濟與效率的考量、這些客戶正不斷擴充、將AI/ML功能移轉至公有雲、混合雲或多雲端雲端部署。但是、它們受到其他基礎架構所能接觸到的資料所約束。NetApp解決方案可解決所需的各種網路安全威脅 "[資料保護](#)" 以及安全性評估、並與Protopia資料轉型結合使用、將影像處理AI/ML工作負載移轉至雲端的相關風險降至最低。

如需其他跨其他產業的邊緣運算和AI推斷使用案例、請參閱 "[TR-4886 AI在邊緣推斷](#)" 以及NetApp AI部落格、"[情報與隱私](#)"。

技術總覽

本節概述完成本解決方案所需的各種技術元件。

Protopia

Protopia AI提供不引人注目的純軟體解決方案、可在當今市場上進行機密推論。Protopia解決方案可將敏感資訊的曝險降至最低、為推論服務提供無與倫比的保護。AI只會提供資料記錄中的資訊、而這些資訊對於執行手邊的工作而言、是真正不可或缺的。大多數推斷工作並不會使用每個資料記錄中的所有資訊。無論您的AI是使用影像、語音、視訊、甚至是結構化的表格式資料、Protopia都能提供您所需的推斷服務。專利的核心技術使用數學上精心策劃的雜訊、以隨機方式轉換資料、並找出特定ML服務不需要的資訊。此解決方案並不會遮罩資料、而是使用精選的隨機雜訊來變更資料呈現方式。

Protopia解決方案會產生將呈現變更為漸層式的靜止最大化方法的問題、此方法仍會保留輸入功能空間中與模型功能相關的資訊。此探索程序會在訓練結束時、以微調通過的方式執行ML模型。通過後會自動產生一組機率分配、低成本的資料轉換會將這些分配的雜訊樣本套用至資料、在將其傳遞至模型進行推斷之前、先將其模糊化。

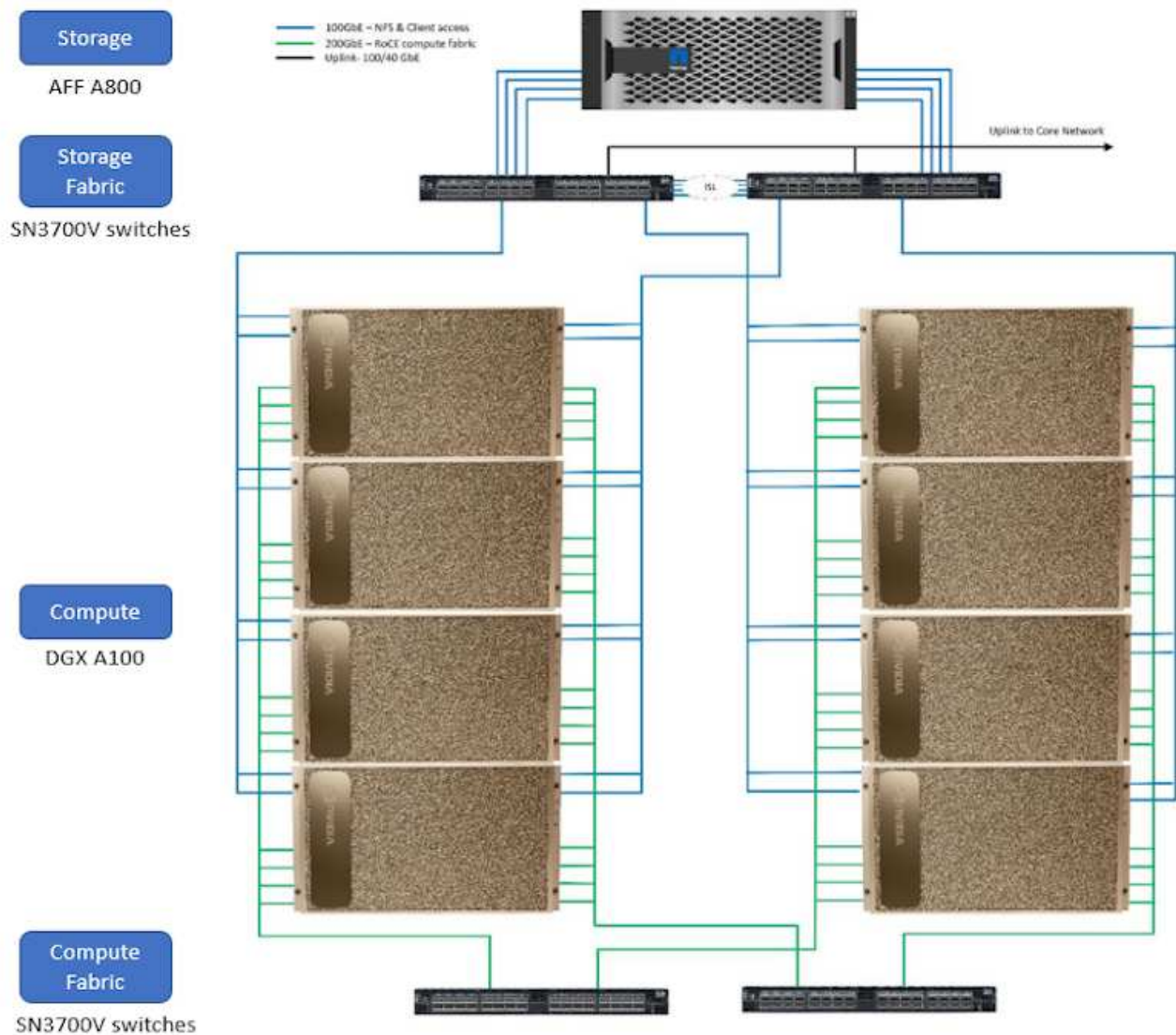
NetApp ONTAP AI

NetApp ONTAP 採用DGX A100系統和NetApp雲端連線儲存系統的NetApp AI參考架構、是由NetApp和NVIDIA開發與驗證的。它為IT組織提供以下優點的架構：

- 消除設計複雜性
- 可獨立擴充運算與儲存設備
- 讓客戶從小規模開始、並無縫擴充
- 提供多種儲存選項、可滿足各種效能與成本效益

AI將DGX A100系統與NetApp的Arde-A800儲存系統與最先進的網路技術緊密整合。ONTAP AFF藉由消除設計複雜度和猜測、AI簡化了AI部署。ONTAP客戶可以從小規模開始、不中斷營運地成長、同時以智慧方式管理從邊緣到核心到雲端及後端的資料。

下圖顯示ONTAP 採用DGX A100系統的整個解決方案系列有多種差異。多達八個DGX A100系統可驗證系統效能。AFF透過將儲存控制器配對新增至ONTAP VMware叢集、架構可擴充至多個機架、以線性效能支援許多DGX A100系統及PB儲存容量。此方法可根據所使用的DL機型大小和所需的效能指標、靈活地改變運算與儲存設備的比率。



如需ONTAP 更多關於AI的資訊、請參閱 ["NVA-1153：採用ONTAP NVIDIA DGX A100系統和Mellanox Spectrum乙太網路交換器的NetApp W人工 智慧。"](#)

NetApp ONTAP

NetApp最新一代的儲存管理軟體《支援產業》（NetApp）、可讓企業將基礎架構現代化、並移轉至雲端就緒的資料中心。ONTAP利用領先業界的資料管理功能ONTAP、無論資料位於何處、只要使用一組工具、即可管理及保護資料。您也可以自由地將資料移至任何需要的位置：邊緣、核心或雲端。包含許多功能、可簡化資料管理、加速及保護關鍵資料、並在混合雲架構中提供新一代基礎架構功能。ONTAP

NetApp DataOps工具套件

NetApp DataOps Toolkit是Python程式庫、讓開發人員、資料科學家、DevOps工程師及資料工程師能夠輕鬆執行各種資料管理工作、例如近乎即時地配置新的資料Volume或JupyterLab工作區、近乎即時地複製資料Volume或JupyterLab工作區、並近乎即時地擷取資料磁碟區或JupyterLab工作區的快照、以供追蹤或建立基準。此Python程式庫可做為命令列公用程式、或是可匯入任何Python程式或Jupyter筆記型電腦的函數庫。

NVIDIA Triton Inference伺服器

NVIDIA Triton Inference Server是開放原始碼的推斷服務軟體、可協助標準化模型部署與執行、在正式作業中提供快速且可擴充的AI。Triton Inference Server可讓團隊從任何GPU或CPU基礎架構上的任何架構、部署、執行及擴充訓練精良的AI模型、進而簡化AI推斷。Triton Inference Server支援所有主要架構、例如TensorFlow、NVIDIA TensorRT、PyTorch、MXNet、OpenVINO等。Triton與Kubernetes整合、提供協調與擴充功能、可用於所有主要公有雲AI和Kubernetes平台。它也與許多MLOps軟體解決方案整合。

PyTorch

"PyTorch" 是開放原始碼ML架構。這是最佳化的張量程式庫、可用於深度學習、使用GPU和CPU。PyTorch套件包含多維度游標的資料結構、可提供許多公用程式、以便在其他實用的公用程式之間有效序列化游標。它也有CUDA對應產品、可讓您在具有運算能力的NVIDIA GPU上執行張量運算。在這項驗證中、我們使用OpenCV-Python (CV2) 程式庫來驗證我們的模型、同時充分運用Python最直覺式的電腦視覺概念。

簡化資料管理

資料管理對於企業IT營運和資料科學家而言至關重要、因此可將適當的資源用於AI應用程式和訓練AI/ML資料集。下列關於NetApp技術的其他資訊超出此驗證範圍、但可能會因您的部署而有所差異。

包含下列功能的資料管理軟體、可簡化及簡化作業、並降低您的總營運成本：ONTAP

- 即時資料精簡與擴充重複資料刪除技術。資料壓縮可減少儲存區塊內的空間浪費、重複資料刪除技術可大幅提升有效容量。這適用於本機儲存的資料、以及分層至雲端的資料。
- 最低、最大及可調適的服務品質 (AQO)。精細的服務品質 (QoS) 控制有助於維持高共享環境中關鍵應用程式的效能等級。
- NetApp FabricPool自動將冷資料分層至公有和私有雲端儲存選項、包括Amazon Web Services (AWS)、Azure和NetApp StorageGRID 等儲存解決方案。如需FabricPool 更多有關資訊、請參閱 ["TR-4598 : FabricPool 最佳實務做法"](#)。

加速並保護資料

提供優異的效能與資料保護、並以下列方式擴充這些功能：ONTAP

- 效能與較低的延遲。以最低的延遲提供最高的處理量。ONTAP
- 資料保護：支援所有平台的通用管理功能、可提供內建的資料保護功能。ONTAP
- NetApp Volume Encryption (NVE)。支援內建和外部金鑰管理、提供原生Volume層級的加密功能。ONTAP
- 多租戶和多因素驗證。支援以最高安全等級共享基礎架構資源。ONTAP

符合未來需求的基礎架構

下列功能可協助滿足嚴苛且不斷變化的業務需求：ONTAP

- 無縫擴充與不中斷營運。支援在不中斷營運的情況下、將容量新增至現有控制器和橫向擴充叢集。ONTAP客戶可以升級至最新技術、例如NVMe和32GB FC、而不需進行昂貴的資料移轉或中斷運作。
- 雲端連線：NetApp是最具雲端連線能力的儲存管理軟體、可在所有公有雲中選擇軟體定義儲存 (AI) 和雲端原生執行個體 (NetApp) ONTAP Select Cloud Volumes Service。
- 與新興應用程式整合。利用支援現有企業應用程式的相同基礎架構、為新一代平台和應用程式提供企業級資料服務、例如自動駕駛車輛、智慧城市和產業4.0。ONTAP

NetApp Astra Control

NetApp Astra產品系列提供儲存與應用程式感知資料管理服務、適用於Kubernetes應用程式的內部部署與公有雲、採用NetApp儲存與資料管理技術。它可讓您輕鬆備份Kubernetes應用程式、將資料移轉至不同的叢集、並即時建立運作中的應用程式複本。如果您需要管理在公有雲上執行的Kubernetes應用程式、請參閱的文件 "[Astra 控制服務](#)"。Astra Control Service是NetApp管理的服務、可在Google Kubernetes Engine（GKE）和Azure Kubernetes Service（KS）中、提供Kubernetes叢集的應用程式感知資料管理功能。

NetApp Astra Trident

Astra "[Trident](#)" NetApp是適用於Docker和Kubernetes的開放原始碼動態儲存協調工具、可簡化持續儲存的建立、管理和使用。Kubernetes原生應用程式Trident直接在Kubernetes叢集內執行。Trident可讓客戶將DL Container映像無縫部署到NetApp儲存設備、並為AI Container部署提供企業級體驗。Kubernetes使用者（ML開發人員、資料科學家等）可以建立、管理及自動化協調與複製、以充分利用NetApp技術所提供的進階資料管理功能。

NetApp BlueXP 複製與同步

"[BlueXP 複製與同步](#)" 是一項NetApp服務、可快速且安全地同步資料。無論您需要在內部部署 NFS 或 SMB 檔案共用之間傳輸檔案、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 複製與同步功能可快速安全地將檔案移至所需的位置。資料傳輸完成後、即可在來源和目標上完全使用。BlueXP 複製和 Sync 會根據預先定義的排程持續同步資料、只移動資料量、將資料複製所花的時間和金錢降到最低。BlueXP 複製與同步是一種軟體即服務（SaaS）工具、設定與使用極為簡單。BlueXP 複製與同步所觸發的資料傳輸是由資料代理人執行。您可以在 AWS、Azure、Google Cloud Platform 或內部部署中部署 BlueXP 複製和同步資料代理人。

NetApp BlueXP 分類

採用強大的AI演算法、"[NetApp BlueXP 分類](#)" 在整個資料產業中提供自動化控管與資料治理功能。您可以輕鬆找出成本節約效益、找出法規遵循與隱私權的考量、並找出最佳化商機。BlueXP 分類儀表板可讓您深入瞭解如何識別重複的資料、以消除備援、對應個人、非個人及敏感資料、並針對敏感資料和異常狀況開啟警示。

測試與驗證計畫

針對此解決方案設計、已驗證下列三種情境：

- JupyterLab工作區中的一項推斷工作、無論是否使用Protopia混淆、都是使用適用於Kubernetes的NetApp DataOps Toolkit進行協調。
- Kubernetes上的批次推斷工作（無論是否有Protopia混淆）、是使用NetApp DataOps Toolkit for Kubernetes所協調的資料Volume。
- 使用NVIDIA Triton Inference Server執行個體的推斷工作、使用適用於Kubernetes的NetApp DataOps Toolkit進行協調。在啟動Triton推斷API之前、我們先將Protopia混淆套用至映像、以模擬透過網路傳輸的任何資料都必須模糊不清的常見需求。此工作流程適用於在信任區域內收集資料、但必須在信任區域外傳遞資料以供推斷的使用案例。如果不使用Protopia混淆、就無法在未將敏感資料離開信任區域的情況下實作此類工作流程。

測試組態

下表概述解決方案設計驗證環境。

元件	版本
Kubernetes	1.21.6
NetApp Astra Trident SCSI驅動程式	22.01.0
適用於Kubernetes的NetApp DataOps工具套件	2.3.0
NVIDIA Triton Inference伺服器	21.11-py3.

測試程序

本節說明完成驗證所需的工作。

先決條件

若要執行本節所述的工作、您必須安裝並設定下列工具、才能存取Linux或MacOS主機：

- Kubectl（設定為存取現有的Kubernetes叢集）
 - 您可以找到安裝與組態指示 ["請按這裡"](#)。
- 適用於Kubernetes的NetApp DataOps工具套件
 - 您可以找到安裝指示 ["請按這裡"](#)。

案例1—JupyterLab的隨需推斷

1. 為AI / ML推斷工作負載建立Kubernetes命名空間。

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

2. 使用NetApp DataOps Toolkit來配置持續磁碟區、以儲存您要在其中執行推斷的資料。

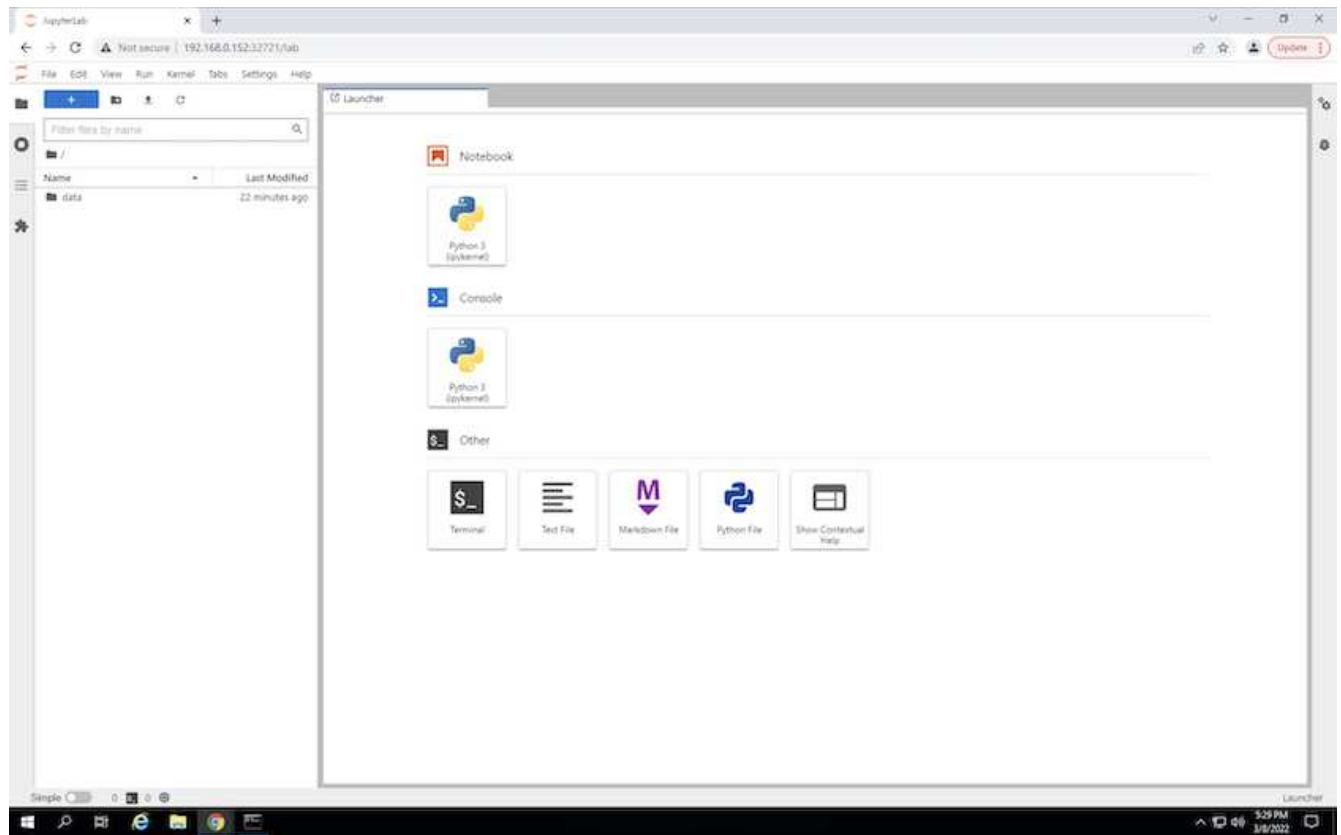
```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully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

3. 使用NetApp DataOps Toolkit建立新的JupyterLab工作區。使用「-mount-PVC」選項來掛載上一步建立的持續磁碟區。使用「-nvidia-GPU」選項、視需要將NVIDIA GPU分配給工作區。

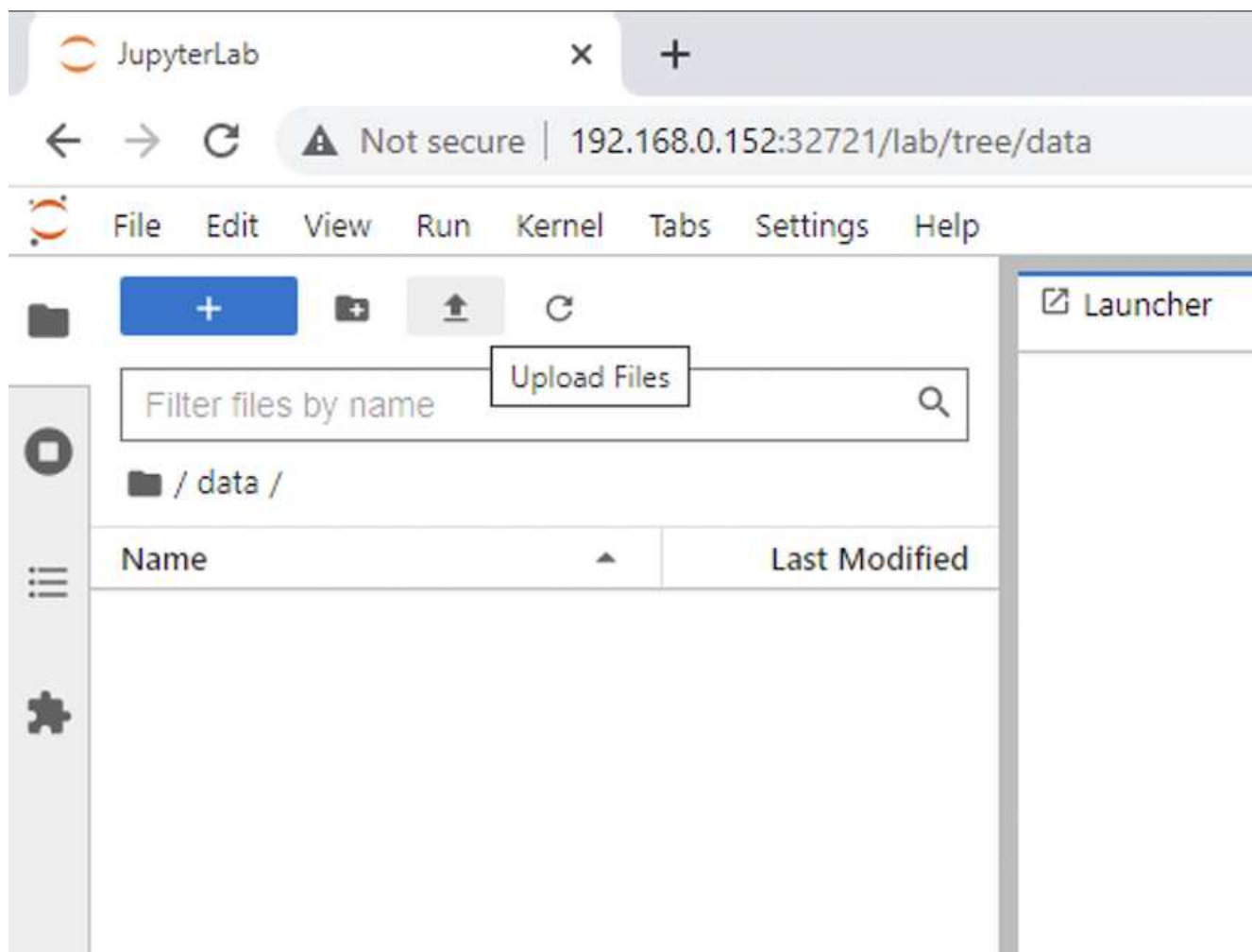
在以下範例中、持續性磁碟區「推斷資料」會掛載到JupyterLab工作區容器、位於「/home/jovyan/data」。使用正式的Project Jupyter Container映像時、「/home/jovyan」會顯示為JupyterLab網路介面中的頂層目錄。

```
$ 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
```

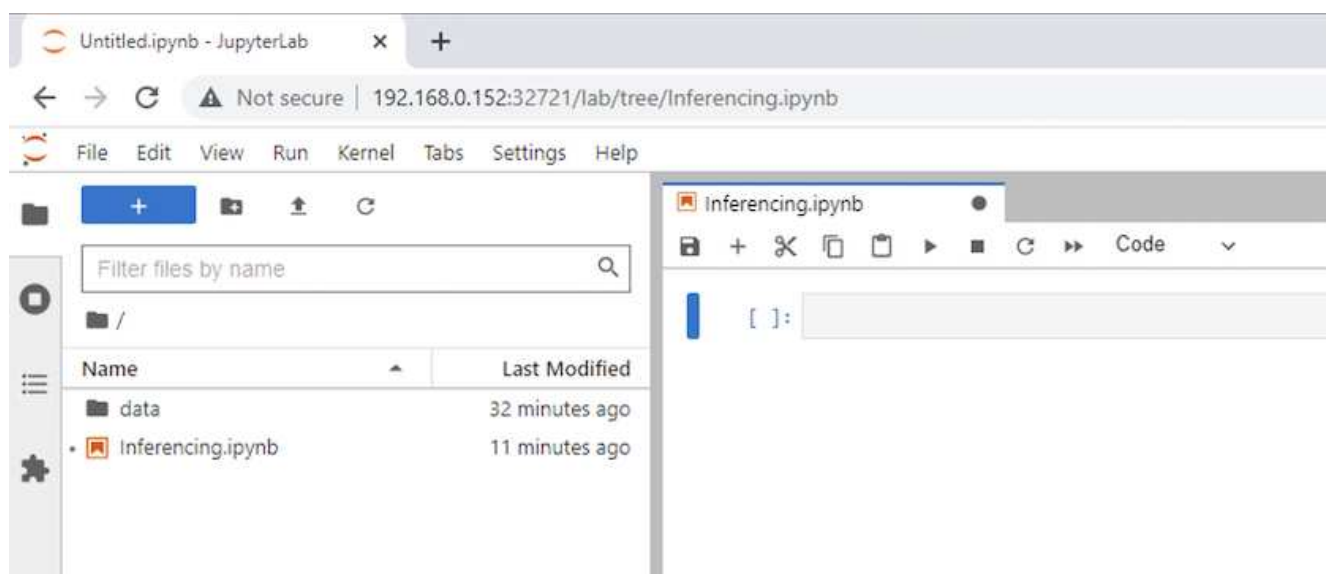
4. 使用「create jupyterlab」命令輸出中指定的URL存取JupyterLab工作區。資料目錄代表掛載到工作區的持續磁碟區。



5. 開啟「DATA」目錄、然後上傳要執行提示的檔案。檔案上傳至資料目錄時、會自動儲存在掛載至工作區的持續磁碟區上。若要上傳檔案、請按一下「上傳檔案」圖示、如下圖所示。



6. 返回最上層目錄並建立新的筆記本。



7. 在筆記本中加入推斷程式碼。下列範例顯示影像偵測使用案例的推斷代碼。

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

STEP 3-1: Clean (Without obfuscation) detection

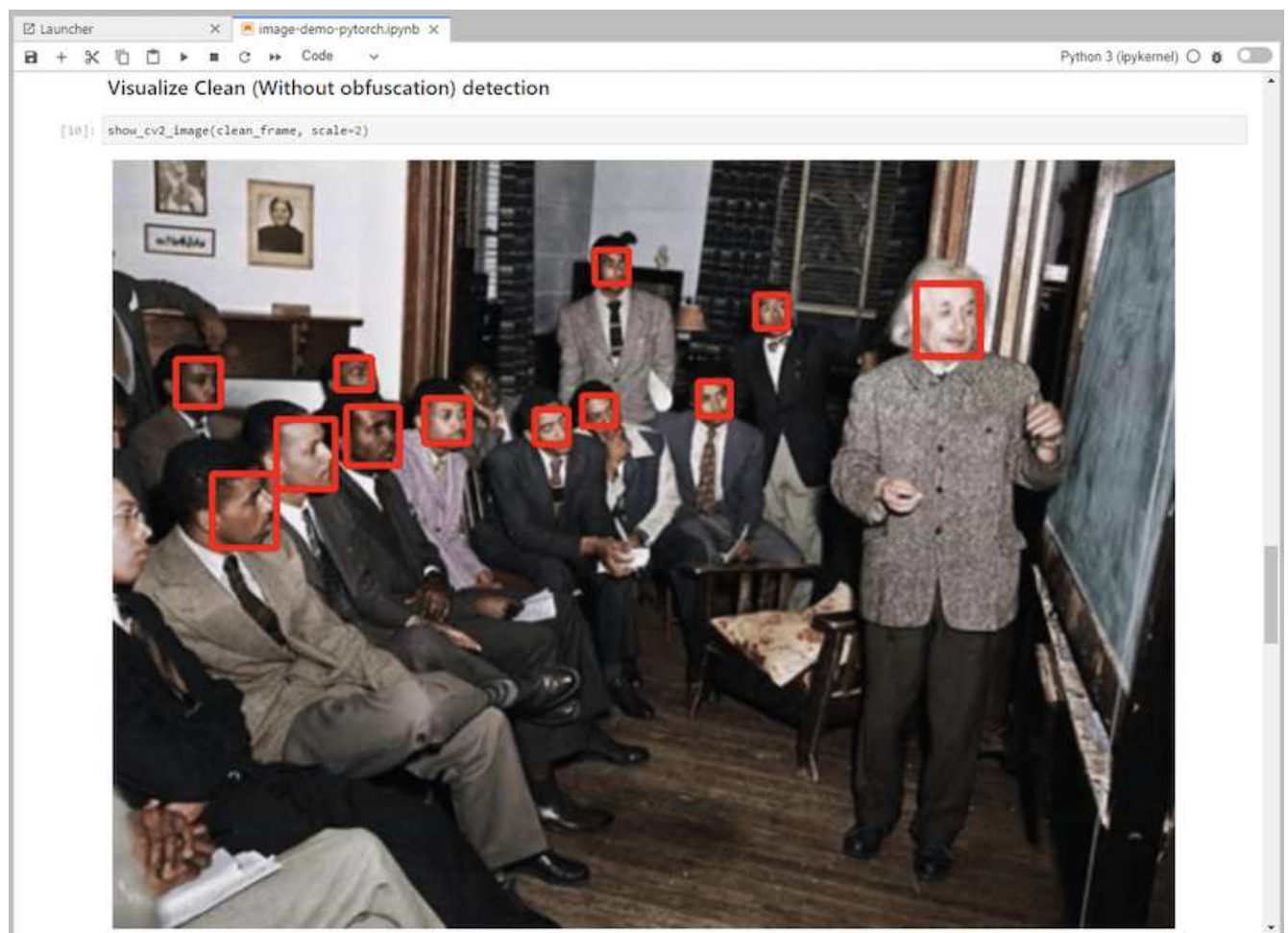
[9]: # get current frame
frame = input_image

# 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. 將Protopia混淆新增至您的推斷程式碼。Protopia直接與客戶合作、提供特定使用案例的文件、並不在本技術報告的範圍之內。以下範例顯示新增Protopia混淆功能時、影像偵測使用案例的推斷程式碼。


```
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

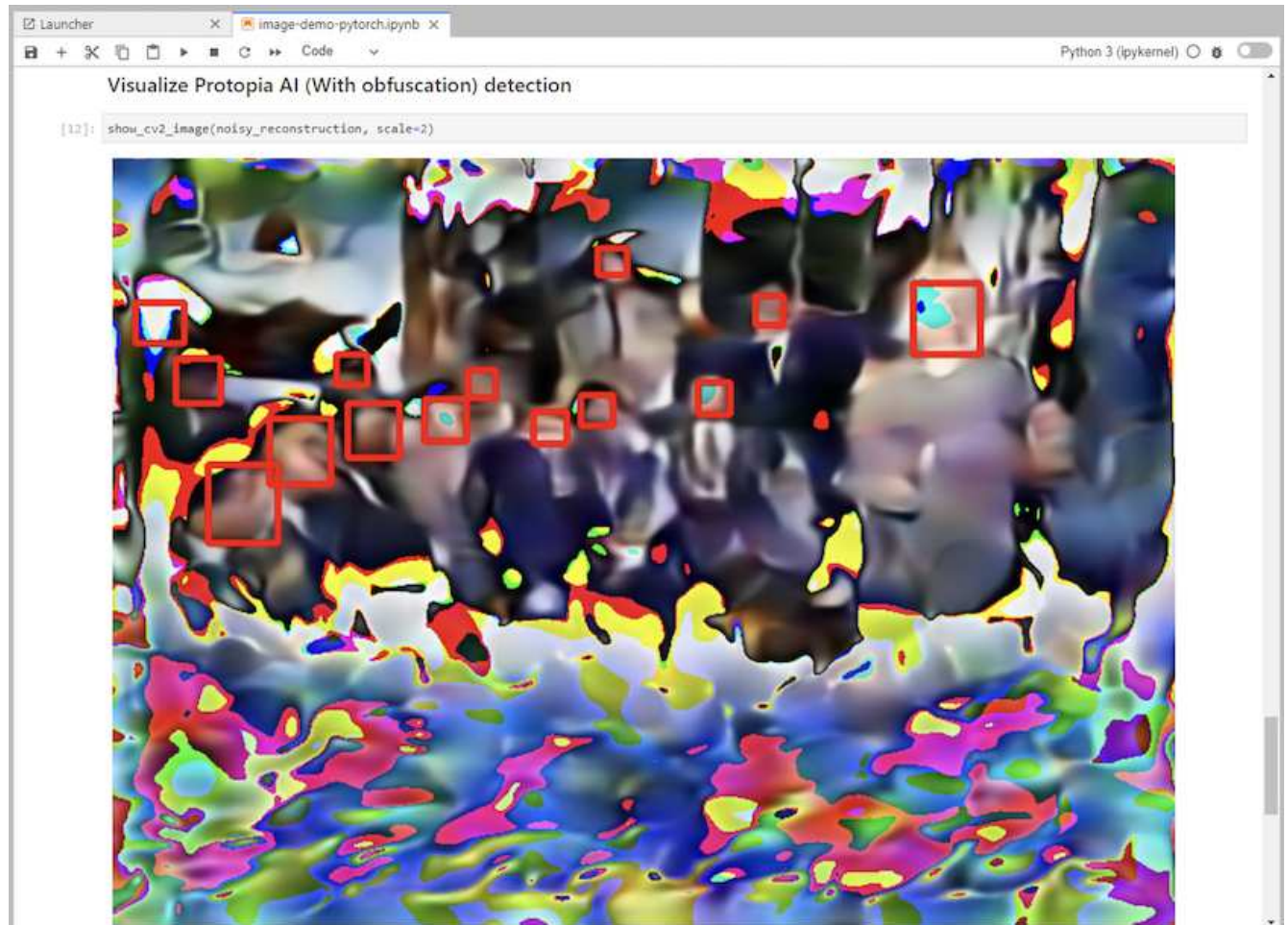
      # preprocess input
      preprocessed_input = preprocess_input(frame)
      preprocessed_input = torch.Tensor(preprocessed_input).to(device)

      # 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)
```



案例2：Kubernetes上的批次推斷

1. 為AI / ML推斷工作負載建立Kubernetes命名空間。

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

2. 使用NetApp DataOps Toolkit來配置持續磁碟區、以儲存您要在其中執行推斷的資料。

```
$ 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. 在新的持續磁碟區中填入您要執行推斷的資料。

有多種方法可將資料載入至PVC。如果您的資料目前儲存在S3相容的物件儲存平台、例如NetApp StorageGRID 功能區或Amazon S3、您就可以使用 ["NetApp DataOps Toolkit S3 Data Mover功能"](#)。另一種簡單的方法是建立JupyterLab工作區、然後透過JupyterLab網頁介面上傳檔案、如「[一節中步驟3至5所述案例1–JupyterLab的隨需推斷](#)。」

4. 為批次推斷工作建立Kubernetes工作。下列範例顯示影像偵測使用案例的批次推斷工作。此工作會在一組映像中的每個映像上執行推斷、並將推斷準確度指標寫入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. 確認推斷工作已成功完成。

```

$ 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

```

- 在推斷工作中加入Protopia混淆。您可以在本技術報告範圍之外的Protopia中、找到直接新增Protopia混淆的使用案例特定指示。下列範例顯示使用0.8的Alpha值新增Protopia模糊處理時、面偵測使用案例的批次推斷工作。此工作會先套用Protopia混淆、再對一組影像中的每個影像進行推斷、然後將推斷準確度指標寫入stdout。

我們重複此步驟以取得Alpha值、包括0.05、0.1、0.2、0.4、0.6、0.8、0.9及0.95。您可以在中看到結果"[「推斷準確度比較」](#)。"

```
$ 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. 確認推斷工作已成功完成。

```
$ 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
```

案例3–NVIDIA Triton Inference Server

1. 為AI / ML推斷工作負載建立Kubernetes命名空間。

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

2. 使用NetApp DataOps Toolkit來配置持續磁碟區、以作為NVIDIA Triton Inference Server的模型儲存庫。

```
$ 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. 將您的模型儲存在中的新持續磁碟區上 **格式** NVIDIA Triton Inference伺服器也能辨識這點。

有多種方法可將資料載入至PVC。簡單的方法是建立JupyterLab工作區、然後透過JupyterLab網路介面上傳檔案、如「」中的步驟3至5所述[案例1–JupyterLab的隨需推斷](#)。」

4. 使用NetApp DataOps Toolkit部署新的NVIDIA Triton Inference Server執行個體。


```
$ 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用戶端SDK執行推斷工作。下列Python程式碼摘錄使用Triton Python用戶端SDK、針對面偵測使用案例執行推斷工作。此範例會呼叫Triton API、並傳入影像以供參考。然後Triton Inference伺服器會收到要求、啟動模型、並傳回推斷輸出、做為API結果的一部分。

```
# 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. 將Protopia混淆新增至您的推斷程式碼。您可以找到直接從Protopia新增Protopia混淆的使用案例特定指示、不過此程序不在本技術報告的範圍之內。以下範例顯示與前述步驟5相同的Python程式碼、但新增了Protopia混淆功能。

請注意、Protopia混淆會套用至映像、然後再傳遞至Triton API。因此、不模糊的影像永遠不會離開本機機器。只有模糊的映像會透過網路傳送。此工作流程適用於在信任區域內收集資料、但需要在信任區域外傳遞資料以進行推斷的使用案例。如果沒有Protopia混淆、就無法在不敏感資料離開信任區域的情況下實作這類工作流程。

```

# 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)

```

推斷準確度比較

為了進行此驗證、我們使用一組原始影像來推斷影像偵測使用案例。然後、我們在相同的影像集上執行相同的推斷工作、並在推斷之前新增Protopia模糊功能。我們使用不同的Alpha值來重複執行Protopia混淆元件的工作。在Protopia混淆的情況下、Alpha值代表套用的模糊處理量、而較高的Alpha值代表較高層級的模糊處理。然後我們比較這些不同路跑的推斷準確度。

下表提供使用案例的詳細資料、並概述結果。

Protopia直接與客戶合作、針對特定使用案例來判斷適當的Alpha值。

元件	詳細資料
模型	FaceBoxes (PyTorch) -
資料集	FDDB資料集

普洛皮亞混淆	Alpha	準確度
否	不適用	0.9337148153739079
是的	0.05	0.9028766627325002
是的	0.1	0.9024301009661478
是的	0.2	0.9081836283186280
是的	0.4	0.90766107482036
是的	0.6	0.8847816568680239
是的	0.8	0.8841195749171925
是的	0.9	0.84554276775252052
是的	0.95	0.84554276775252052

模糊化速度

在這項驗證中、我們將Protopia混淆套用至1920 x 1080像素映像五倍、並測量每次執行模糊化步驟所需的時間量。

我們使用在單一NVIDIA V100 GPU上執行的PyTorch來套用混淆、並在執行期間清除GPU快取。在這五次路跑中、混淆步驟分別花費5.47毫秒、5.27毫秒、4.54毫秒、5.24毫秒和4.84毫秒來完成。平均速度為5.072ms。

結論

資料存在三種狀態：閒置、傳輸和運算。任何AI推斷服務的重要一環、都應該是在整個流程中保護資料免受威脅。在推斷過程中保護資料是非常重要的、因為此程序可能會揭露外部客戶和提供推斷服務的企業的私人資訊。Protopia AI是專為機密AI提供的不突兀軟體專屬解決方案、可在當今市場中提供相關資訊。有了Protopia、AI只會在資料記錄中輸入經過轉換的資訊、而這些資訊對於執行手邊的AI/ML工作而言非常重要、而且不再需要其他資訊。這種隨機轉型並不是一種遮罩形式、而是以數學方式使用「精選雜訊」來變更資料的呈現方式為基礎。

NetApp儲存系統ONTAP 具備各種功能、可提供與本機SSD儲存設備相同或更高的效能、並搭配NetApp DataOps Toolkit、為資料科學家、資料工程師、AI/ML開發人員、企業或企業IT決策者提供下列效益：

- 輕鬆在AI系統、分析和其他關鍵業務系統之間共享資料。這種資料共享可降低基礎架構的負荷、改善效能、並簡化整個企業的資料管理。
- 獨立擴充的運算與儲存設備、可將成本降至最低、並改善資源使用率。
- 利用整合式Snapshot複本與複製、簡化開發與部署工作流程、提供即時且節省空間的使用者工作區、整合式版本控制、以及自動化部署。
- 企業級的資料保護與資料治理功能、可滿足災難恢復、營運不中斷及法規要求。
- 簡化資料管理作業的叫用作業；快速取得資料科學家工作區的Snapshot複本、以便從Jupyter筆記型電腦的NetApp DataOps Toolkit進行備份與追蹤。

NetApp與Protopia解決方案提供靈活的橫向擴充架構、是企業級AI推斷部署的理想選擇。它能保護資料、並為敏感資訊提供隱私保護、在內部部署和混合雲部署中、均能以負責的AI實務做法來滿足機密AI推斷要求。

何處可以找到其他資訊和認可

若要深入瞭解本文件所述資訊、請參閱下列文件和/或網站：

- NetApp ONTAP 數據管理軟體ONTAP —資訊庫

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

- NetApp容器持續儲存設備：NetApp Trident

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

- NetApp DataOps工具套件

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

- NetApp容器持續儲存設備：NetApp Astra Trident

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

- Protopia AI—機密推論

["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 複製與同步

["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 Inference伺服器

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

- NVIDIA Triton Inference Server文件

["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)

感謝

- NetApp首席產品經理Mark Cates
- NetApp技術行銷工程師Sufian Ahmad
- Protopia AI技術長暨教授HADI Esmailzadeh

運用**NetApp AI**進行情緒分析

TR-4910：客戶與NetApp AI溝通時的意見分析

Rick Huang、Sathish Thyagarajan和David Arettette、NetApp Diego Sosa-COSA、SFL Scientific

本技術報告提供設計指南、協助客戶在企業級的全球支援中心、使用NetApp資料管理技術搭配NVIDIA軟體架構、使用傳輸學習和對話式AI來執行情緒分析。此解決方案適用於任何想要從記錄的語音或文字檔案中獲得客戶洞見的產業、這些檔案代表聊天記錄、電子郵件及其他文字或音訊通訊。我們實作了端點對端點傳輸途徑、在採用NetApp雲端連線的All Flash儲存設備的GPU加速運算叢集上、展示自動語音辨識、即時情緒分析、以及深度學習的自然語言處理模式重新訓練功能。您可以訓練並最佳化大量最先進的語言模式、以便與全球支援中心一起快速進行推論、以創造卓越的客戶體驗、以及客觀的長期員工績效評估。

情緒分析是自然語言處理（NLP）中的研究領域、可從文字中擷取正面、負面或中立的情緒。隨著越來越多人來與AI系統互動、對話式AI系統的整合程度已提升到接近全球的程度。情緒分析有多種使用案例、從判斷支援中心員工與來電者對話的績效、以及提供適當的自動回應、到根據公司代表與每季獲利拜訪對象之間的互動、預測公司的股票價格。此外、情緒分析可用來判斷客戶對於品牌所提供產品、服務或支援的看法。

此端點對端點解決方案使用NLP模型來執行高層級的意見分析、以實現支援中心的分析架構。錄音會以書面文字處理、並從對話中的每個句子擷取內容。彙總到儀表板的結果、可用來分析過去和即時的對話情緒。此解決方案可通用化為其他具有類似資料模式和輸出需求的解決方案。有了適當的資料、就能完成其他使用案例。例如、可以使用相同的端點對端點管道來分析公司的營收通話。其他形式的NLP分析、例如主題建模和命名實體辨識（NER）、也可能是因為管道的靈活特性。

NVIDIA Riva、NVIDIA TAO工具套件和NetApp DataOps工具套件共同實現了這些AI實作。NVIDIA的工具可用於使用預先建置的機型和管線、快速部署高效能的AI解決方案。NetApp DataOps Toolkit可簡化各種資料管理工作、加速開發。

客戶價值

企業可以從員工評估工具和客戶反應工具中、看到文字、音訊和視訊對話的價值、以便進行意見分析。主管可從儀表板中所顯示的資訊獲益、以便根據對話的雙方、對員工和客戶滿意度進行評估。

此外、NetApp DataOps Toolkit還能管理客戶基礎架構內的資料版本設定與分配。如此一來、儀表板內的分析功能就會頻繁更新、而不會產生難以管理的資料儲存成本。

使用案例

由於這些支援中心處理的通話數眾多、如果手動執行、則評估通話效能可能需要相當長的時間。傳統方法、例如字彙數和其他方法、可以實現某種自動化、但這些方法無法擷取更細微的動態語言內容和語意。AI建模技術可用於以自動化方式執行這些更細微的分析。此外、隨著NVIDIA、AWS、Google等廠商所發佈的最先進預訓練模型工具、現在可以相對輕鬆地備份和自訂具有複雜模型的端點對端點管線。

支援中心意見分析的端點對端管道、可在員工與來電者交談時即時擷取音訊檔案。然後、這些音訊檔案會被處理成語音對文字元件、以將其轉換成文字格式。對話中的每一句話都會貼上標籤、指出感受（正面、負面或中立）。

情緒分析可為對話提供重要的層面、以評估通話效能。這些想法讓員工與來電者之間的互動更加深入。AI協助的意見儀表板可讓經理即時追蹤對話中的意見、並回顧分析員工過去的通話情況。

有許多預先建置的工具可以以強大的方式結合、以快速建立端點對端點AI管線來解決此問題。在此案例中、NVIDIA Riva程式庫可用於執行兩項系列工作：音訊複本和意見分析。第一種是受監控的學習訊號處理演算法、第二種是受監控的學習NLP分類演算法。這些隨裝即用的演算法可利用NVIDIA TAO工具套件、針對任何相關的商業相關資料使用案例進行微調。如此一來、我們就能針對成本與資源的一小部分、打造出更準確且功能更強大的解決方案。客戶可以整合 "NVIDIA Maxine" 支援中心設計中的GPU加速視訊會議應用程式架構。

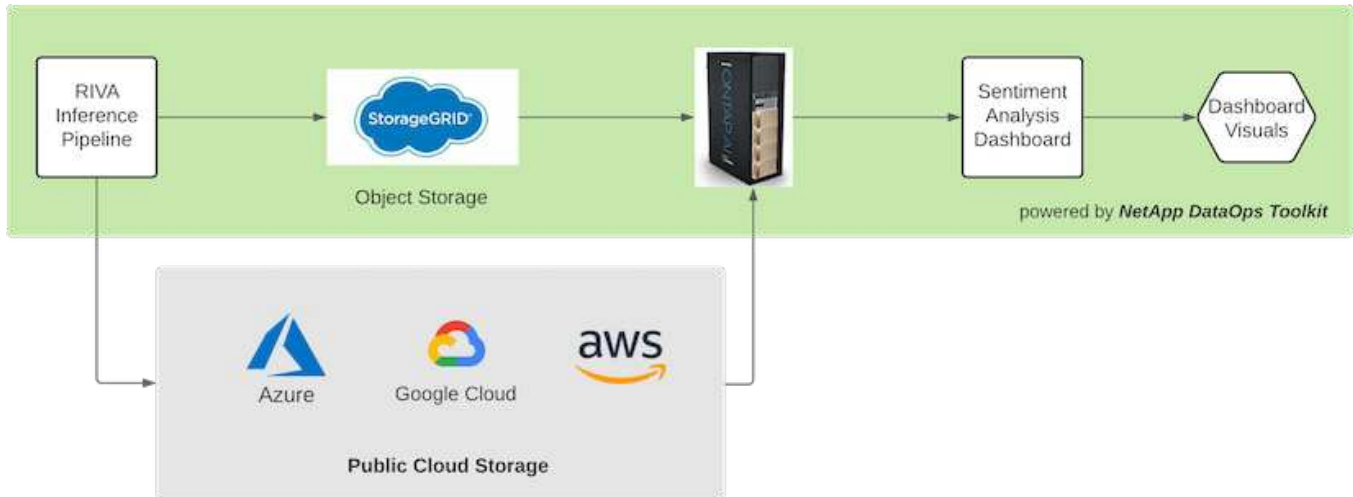
本解決方案的核心是下列使用案例。這兩種使用案例都使用TAO工具套件進行模型微調、而Riva則用於模型部署。

- 語音對文字
- 情緒分析

為了分析員工與客戶之間的支援中心互動、每次客戶對話都可以透過管道進行音訊通話、藉此擷取句子層級的感受。然後、人類可以驗證這些想法、以證明這些想法是正確的、或是視需要加以調整。接著、標記的資料會傳送到微調步驟、以改善情緒預測。如果已存在標記的情緒資料、則可加速模型微調。無論是哪一種情況、管道都可通用於其他需要擷取音訊和分類句子的解決方案。



AI認列輸出會上傳至外部雲端資料庫或公司管理的儲存系統。這類結果會從這個較大的資料庫傳輸到本機儲存設備、以便在儀表板內使用、以便為經理顯示意見分析。儀表板的主要功能是即時與客戶服務員工溝通。經理可在電話中即時更新每句話的感受、以及對員工過去績效或客戶反應的歷史審查、以評估並提供員工意見回饋。



。"NetApp DataOps工具套件" 即使在Riva推論管道產生「氣氛」標籤之後、仍可繼續管理資料儲存系統。這些AI結果可上傳至由NetApp DataOps Toolkit管理的資料儲存系統。資料儲存系統必須能夠管理數百個插件、並每分鐘選取一次。本機裝置儲存系統會即時查詢較大的資料儲存設備以供擷取。您也可以查詢較大的資料儲存執行個體、以取得歷史資料、進一步提升儀表板體驗。NetApp DataOps工具套件可快速複製資料、並在所有使用資料的儀表板上發佈資料、藉此協助這兩種用途。

目標對象

解決方案的目標對象包括下列群組：

- 員工經理
- 資料工程師/資料科學家
- IT管理員（內部部署、雲端或混合式）

追蹤對話過程中的情緒、是評估員工績效的寶貴工具。經理可以使用AI儀表板、即時查看員工和來電者的感受如何改變、以便進行即時評估和指導課程。此外、企業也能從參與聲音對話、文字聊天機器人程式和視訊會議的客戶身上獲得寶貴的客戶見解。這類客戶分析使用現代化、最先進AI模式和工作流程的大規模多式模式處理功能。

在資料方面、支援中心每天都會處理大量的音訊檔案。NetApp DataOps工具套件可協助您定期微調模型和意見分析儀表板、以處理資料。

IT管理員也能從NetApp DataOps Toolkit獲益、因為它可讓他們在部署與正式作業環境之間快速搬移資料。NVIDIA環境和伺服器也必須加以管理和散佈、才能即時推斷。

架構

此支援中心解決方案的架構以NVIDIA預先建置的工具和NetApp DataOps Toolkit為核心。NVIDIA的工具可用於使用預先建置的機型和管線、快速部署高效能AI解決方案。NetApp DataOps Toolkit可簡化各種資料管理工作、加速開發。

解決方案技術

"NVIDIA Riva" 是GPU加速的SDK、可用於建置多式對話AI應用程式、在GPU上提供即時效能。NVIDIA訓練、調適及最佳化（TAO）工具套件提供更快速、更簡單的方法來加速訓練、並快速建立高度準確且效能優異的網域

專屬AI模型。

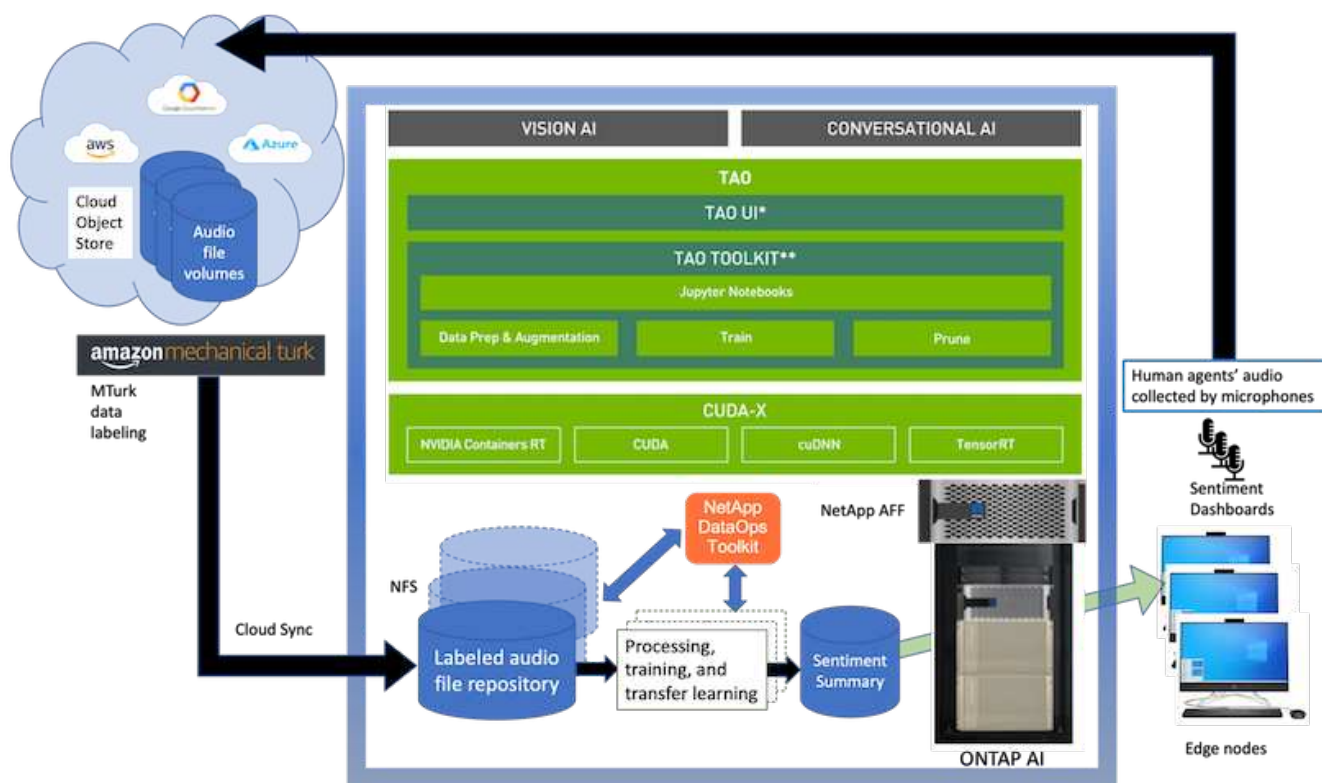
NetApp DataOps Toolkit是Python程式庫、可讓開發人員、資料科學家、DevOps工程師及資料工程師輕鬆執行各種資料管理工作。這包括近乎即時地配置新的資料磁碟區或JupyterLab工作區、近乎即時地複製資料磁碟區或JupyterLab工作區、以及近乎即時的資料磁碟區快照或JupyterLab工作區、以供追蹤及建立基準。

架構圖

下圖顯示解決方案架構。有三大環境類別：雲端、核心和邊緣。每個類別都可以分散在不同的地理位置。例如、雲端的物件儲存區在不同區域的儲存區中、音訊檔案位於儲存區中、而核心則可能包含透過高速網路或 NetApp BlueXP 複製與同步連結的資料中心。邊緣節點表示個人人力代理商的日常工作平台、其中提供互動式儀表板工具和麥克風、可視覺化感受、並從與客戶的對話中收集音訊資料。

在GPU加速的資料中心中、企業可以使用NVIDIA "Riva" 構建對話式AI應用程式的架構 "TAO工具套件" 運用轉介L型學習技術、連接模型的微調和再訓練。這些運算應用程式和工作流程均由提供支援 "NetApp DataOps工具套件"、實現ONTAP 最佳的資料管理功能。此工具組可讓企業資料團隊透過快照和複本、快速建立模型原型、並建立相關的結構化和非結構化資料、以利追蹤、版本管理和A/B測試、進而提供安全性、治理、以及法規遵循。請參閱一節 "儲存設計" 以取得更多詳細資料。

本解決方案說明音訊檔案處理、NLP模式訓練、傳輸學習及資料管理詳細步驟。最終端點對端點的傳輸途徑會產生即時顯示在人力支援代理程式儀表板上的意見摘要。



硬體需求

下表列出實作解決方案所需的硬體元件。在解決方案的任何特定實作中使用的硬體元件、可能會因客戶需求而異。

回應延遲測試	時間（毫秒）
資料處理	10.
推斷	10.

這些回應時間測試是針對560個對話中超過50、000個音訊檔案執行。每個音訊檔案的大小約為100 KB、轉換成wav時約為1 MB。資料處理步驟會將mp3轉換成wav檔案。推斷步驟會將音訊檔案轉換成文字、並從文字擷取內容。這些步驟彼此獨立、可平行化以加速程序。

考慮到在各直營店之間傳輸資料的延遲、經理應能在句子結尾的一秒內看到即時情緒分析的更新。

NVIDIA Riva硬體

硬體	需求
作業系統	Linux x86_64
GPU記憶體（ASR）	串流模式：約5600 MB非串流機型：約3100 MB
GPU記憶體（NLP）	每個Bert機型約500MB

NVIDIA TAO工具套件硬體

硬體	需求
系統RAM	32GB
GPU RAM	32GB
CPU	8核心
GPU	NVIDIA（A100、V100和RTX 30x0）
SSD	100GB

Flash儲存系統

NetApp ONTAP 產品9.

NetApp最新一代的儲存管理軟體- NetApp 9.9、可讓企業將基礎架構現代化、並移轉至雲端就緒的資料中心。ONTAP利用領先業界的資料管理功能ONTAP、無論資料位於何處、只要使用一組工具、即可管理及保護資料。您也可以自由地將資料移至任何需要的位置：邊緣、核心或雲端。支援眾多功能、可簡化資料管理、加速及保護關鍵資料、並在混合雲架構中提供新一代基礎架構功能。ONTAP

NetApp BlueXP 複製與同步

"BlueXP 複製與同步" 是一項NetApp服務、可讓您在內部部署NFS或SMB檔案共用之間傳輸檔案至下列任一目標、以實現快速且安全的資料同步：

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

- Amazon簡易儲存服務 (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Azure Blob
- Google Cloud Storage
- IBM Cloud 物件儲存設備

BlueXP 複製與同步功能可快速安全地將檔案移至所需的位置。資料傳輸完成後、即可在來源和目標上完全使用。BlueXP 複製與同步功能會根據預先定義的排程、持續同步資料、只移動資料量、將資料複製所花費的時間與金錢減至最低。BlueXP 複製與同步是一種軟體即服務 (SaaS) 工具、易於設定與使用。BlueXP 複製與同步所觸發的資料傳輸是由資料代理人執行。您可以在 AWS 、 Azure 、 Google Cloud Platform 或內部部署中部署 BlueXP 複製和同步資料代理人。

NetApp StorageGRID

由軟體定義的物件儲存套件可無縫支援各種公有、私有及混合式多雲端環境的使用案例。StorageGRIDNetApp StorageGRID 支援領先業界的創新技術、可儲存、保護及保留非結構化資料、以供多種用途使用、包括長期的自動化生命週期管理。如需詳細資訊、請參閱 "[NetApp StorageGRID](#)" 網站。

軟體需求

下表列出實作此解決方案所需的軟體元件。在解決方案的任何特定實作中使用的軟體元件、可能會因客戶需求而異。

主機	需求
Riva (前身為JARVIS)	1.4.0
TAO工具套件 (前身為TransferLearning Toolkit)	3.0
ONTAP	9.9.1
DGX OS	5.1
DOTK	2.0.00.0

NVIDIA Riva軟體

軟體	需求
Docker	>19.02 (安裝NVIDIA泊塢視窗) >=19.03 (若未使用DGX)
NVIDIA驅動程式	465.19.01 + 418.40 + 、440.33 + 、450.51 + 、460.27 + (適用於資料中心GPU)
Container作業系統	Ubuntu 20.04
CUDA	11.3.0
cublas	11.5.1.101.
CUDNN	8.2.0.41
NCCL	2.9.6
TensorRT	7.2.3.4

軟體	需求
Triton Inference伺服器	2.9.0

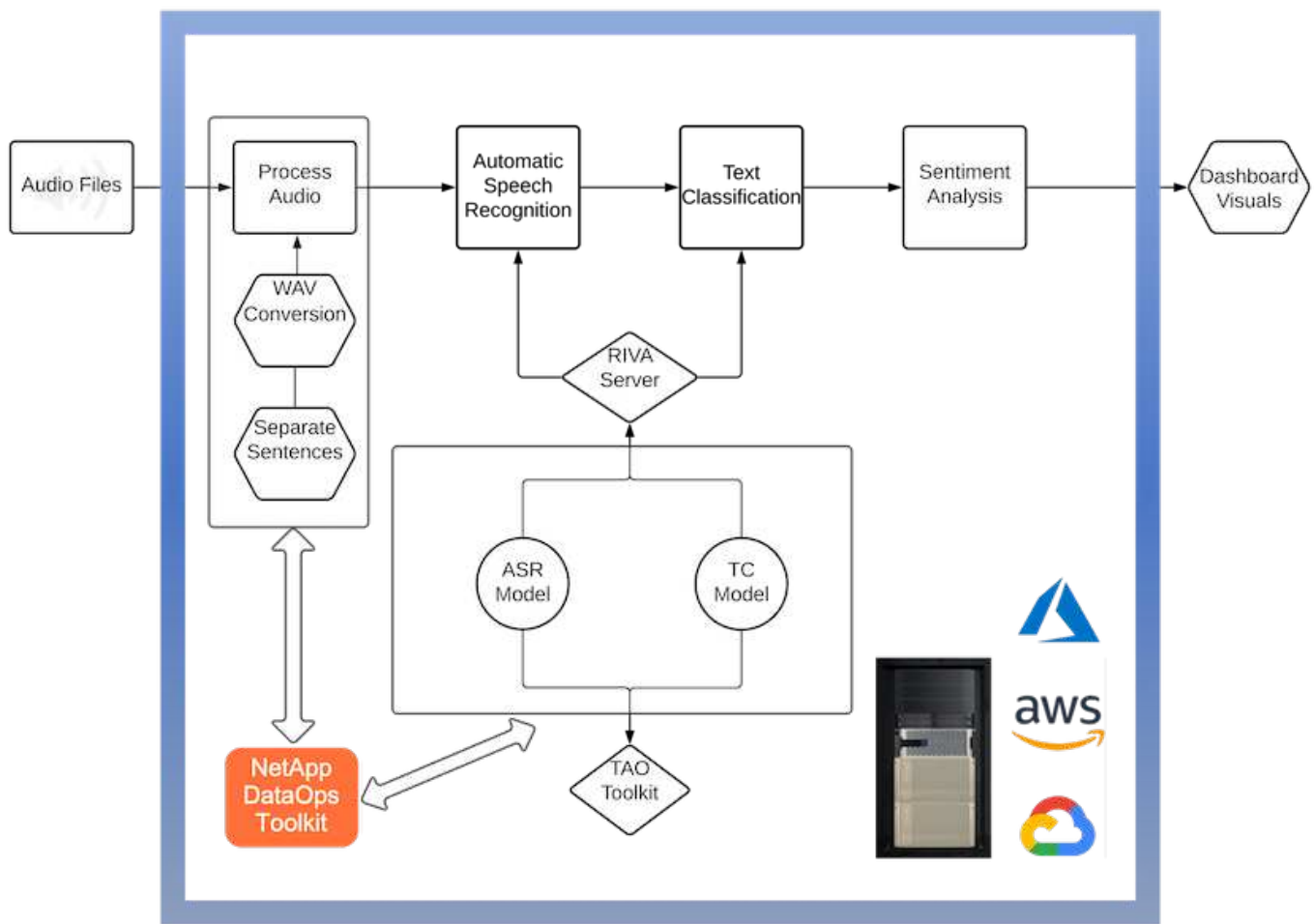
NVIDIA TAO Toolkit軟體

軟體	需求
Ubuntu 18.04 LTS	18.04
Python	>=3.6.9
Docker	>19.03.5
Docker API	1.40
nvidia-container工具套件	>1.3.0-1
nvidia-container執行時間	3.4.0-1
nvidia-docker2	2.5.0-1
NVIDIA驅動程式	>455
Python-pip	>21.06.
nvidia-pyindex	最新版本

使用案例詳細資料

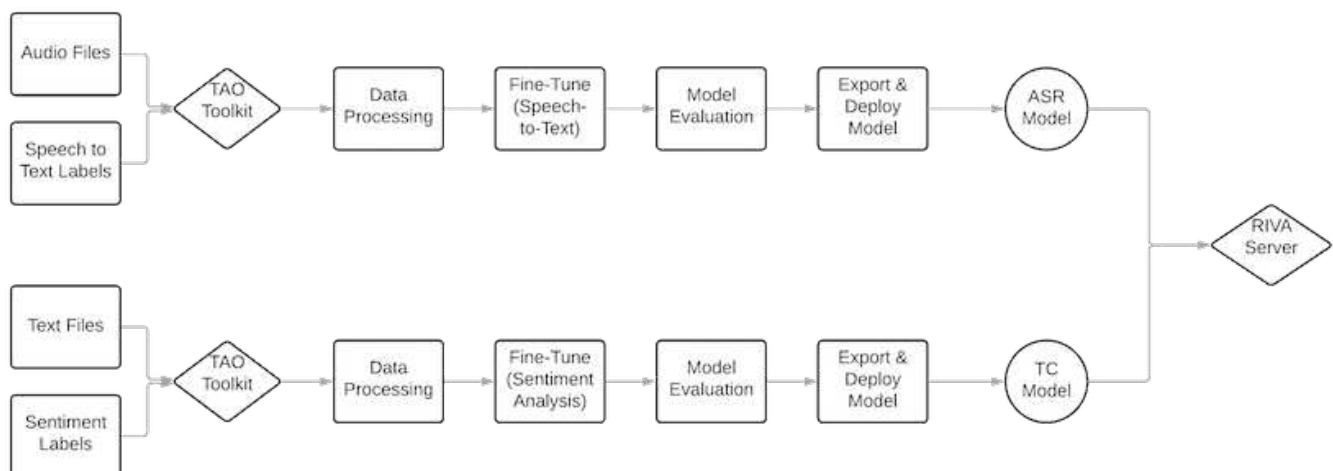
本解決方案適用於下列使用案例：

- 語音對文字
- 情緒分析



語音對文字的使用案例、是從擷取支援中心的音訊檔案開始。然後處理此音訊、以符合Riva所需的結構。如果音訊檔案尚未分割成分析單位、則必須先將音訊傳送至Riva。音訊檔案處理完畢後、會以API呼叫的形式傳送至Riva伺服器。伺服器採用其託管的眾多機型之一、並傳回回應。此語音對文字（自動語音辨識的一部分）會傳回音訊的文字呈現。之後、管線會切換至「意見分析」部分。

對於情緒分析、自動語音辨識的文字輸出可做為文字分類的輸入。文字分類是NVIDIA元件、可將文字分類為任何類別。支援中心對話的感受類別從正面到負面、您可以使用套管套件來評估模型的效能、以判斷微調步驟是否成功。



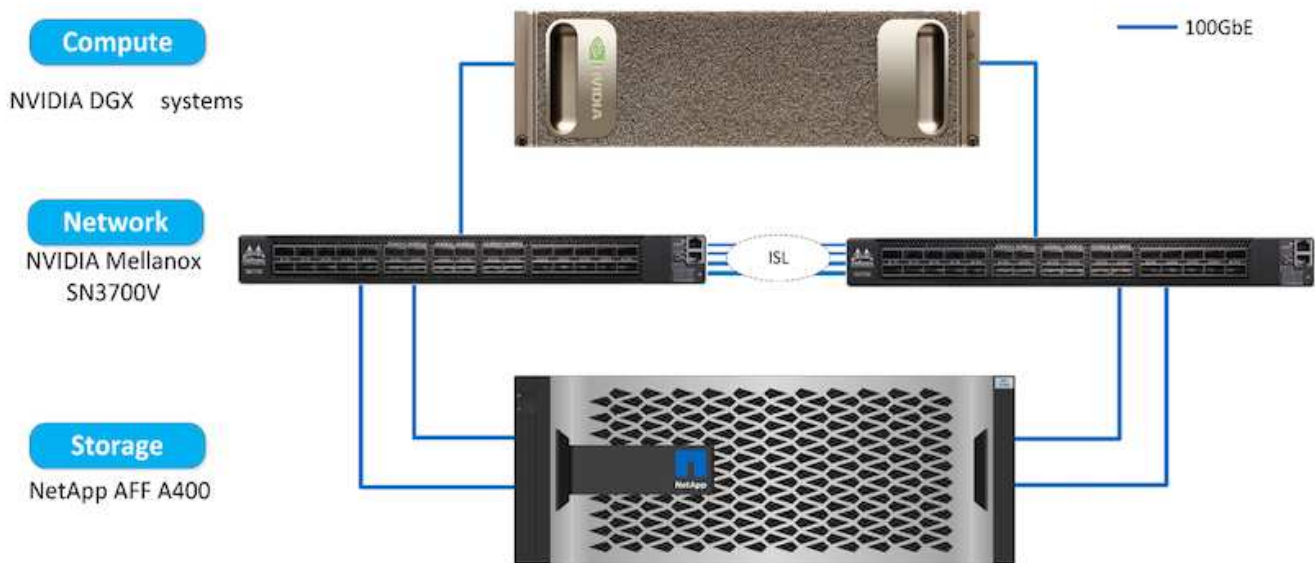
TAO工具套件中的語音對文字和情緒分析也使用類似的管道。主要差異在於使用標籤來微調模型。TAO工具套件管道從資料檔案的處理開始。然後是預先訓練的模型（來自 "[NVIDIA NGC目錄](#)"）使用支援中心資料進行微調。系統會根據其對應的效能指標來評估微調模型、如果它們的效能比預先訓練的模型更高、則會部署到Riva伺服器。

設計考量

本節說明本解決方案不同元件的設計考量。

網路與運算設計

視資料安全性的限制而定、所有資料都必須保留在客戶的基礎架構或安全環境中。



儲存設計

NetApp DataOps Toolkit是管理儲存系統的主要服務。DataOps Toolkit是Python程式庫、讓開發人員、資料科學家、DevOps工程師和資料工程師能夠輕鬆執行各種資料管理工作、例如近乎即時地配置新的資料Volume或JupyterLab工作區、近乎即時地複製資料Volume或JupyterLab工作區、以及近乎即時的資料磁碟區快照或JupyterLab工作區快照、以提供可追蹤性或基準化。此Python程式庫可做為命令列公用程式或可匯入任何Python程式或Jupyter Notebook的函數庫。

Riva最佳實務做法

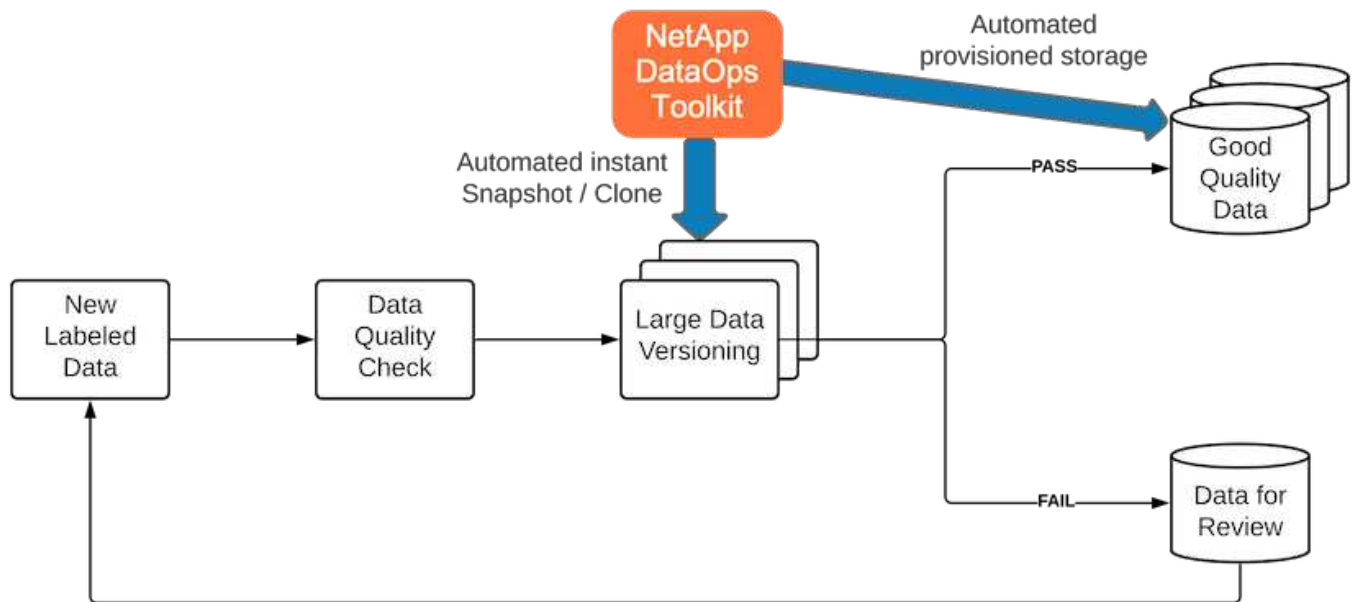
NVIDIA提供多項一般功能 "[最佳資料實務做法](#)" 若使用Riva：

- *如果可能、請使用無損音訊格式。*使用失真轉碼器（例如mp3）可能會降低品質。
- *強化訓練資料。*在音訊訓練資料中加入背景雜訊、一開始就能降低準確度、同時提高健全度。
- *如果使用剪貼的文字、請限制詞彙大小。*許多線上來源包含錯字或輔助詞彙、以及不常見的詞彙。移除這些項目可改善語言模式。
- *如果可能、請使用最低16Khz取樣率。*不過、請盡量不要重新取樣、因為這樣會降低音訊品質。

除了這些最佳實務做法之外、客戶還必須優先收集代表範例資料集、並針對管路的每個步驟提供正確的標籤。換

句話說、範例資料集應按比例反映目標資料集中所示的特定特性。同樣地、資料集註釋者也有責任在準確度和標籤速度之間取得平衡、以使資料的品質和數量達到最大化。例如、此支援中心解決方案需要音訊檔案、標示文字和內容標記。此解決方案的連續性意味著從管線一開始的錯誤會一直傳播到最後如果音訊檔案品質不佳、文字轉譯和翻譯意指標籤也會一樣。

此錯誤傳播同樣適用於受過此資料訓練的模型。如果情緒預測是100%準確、但語音對文字模式的效能不佳、則最終管道將受到初始音訊對文字轉譯的限制。開發人員必須個別考量每種模式的效能、並將其視為較大型管線的元件。在這個特定案例中、最終目標是開發可準確預測感受的管道。因此、評估管道的整體指標是感受的準確度、而這種觀點會直接影響到語音對文字的複本。



NetApp DataOps Toolkit透過近乎即時的資料複製技術來輔助資料品質檢查管道。每個標示的檔案都必須進行評估、並與現有標示的檔案進行比較。在各種資料儲存系統之間散佈這些品質檢查、可確保這些檢查作業快速且有效率地執行。

部署支援中心意見分析

部署解決方案包括下列元件：

1. NetApp DataOps工具套件
2. NGC組態
3. NVIDIA Riva伺服器
4. NVIDIA TAO工具套件
5. 將TAO模型匯出至Riva

若要執行部署、請完成下列步驟：

NetApp DataOps Toolkit：支援中心意見分析

以使用 "NetApp DataOps工具套件"、完成下列步驟：

1. 子母畫面安裝工具套件。

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

2. 設定資料管理

```
netapp_dataops_cli.py config
```

NGC組態：支援中心意見分析

以設定 "NVIDIA NGC"、完成下列步驟：

1. 下載NGC。

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

2. 將目前目錄新增至路徑。

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

3. 您必須設定NGC CLI以供使用、才能執行命令。輸入下列命令、並在出現提示時輸入您的API金鑰。

```
ngc config set
```

若為非Linux型作業系統、請造訪 ["請按這裡"](#)。

NVIDIA Riva伺服器：支援中心意見分析

以設定 "NVIDIA Riva"、完成下列步驟：

1. 從NGC下載Riva檔案。

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

2. 初始化Riva設定 ('Riva_init.sh')。
3. 啟動Riva伺服器 ('Riva_start.sh')。

4. 啟動Riva用戶端 ('Riva_start_client.sh') 。
5. 在Riva用戶端中、安裝音訊處理程式庫 ("FFmpeg")

```
apt-get install ffmpeg
```

6. 啟動 "Jupyter" 伺服器：
7. 執行Riva Inference Pipeline Notebook 。

NVIDIA TAO工具套件：支援中心意見分析

若要設定NVIDIA TAO工具套件、請完成下列步驟：

1. 準備並啟動 "虛擬化環境" 適用於TAO工具套件。
2. 安裝 "必要的套件"。
3. 手動拉出訓練和微調期間使用的影像。

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

4. 啟動 "Jupyter" 伺服器：
5. 執行TAO微調筆記型電腦。

將TAO模式匯出至Riva：支援中心情緒分析

以供使用 "TAO工具套件模組：Riva"、完成下列步驟：

1. 儲存TAO微調筆記型電腦的機型。
2. 將受過訓練的TAO模型複製到Riva模型目錄。
3. 啟動Riva伺服器 (「Riva_start.sh」) 。

部署障礙

以下是您開發自己解決方案時必須謹記的幾點事項：

- 首先安裝NetApp DataOps Toolkit、以確保資料儲存系統以最佳方式執行。
- 必須先安裝NVIDIA NGC、才能驗證影像和機型的下載。
- Riva必須在TAO工具套件之前安裝。Riva安裝會將Docker精靈設定為視需要擷取影像。
- DGX和Docker必須能夠存取網際網路才能下載模型。

驗證結果

如前一節所述、只要有兩個或多個機器學習模式依序執行、錯誤就會傳播到整個管線中。對於此解決方案、句子的意涵是衡量公司股票風險等級的最重要因素。語音對文字模式雖

然對管道至關重要、但在預測意見之前、仍是預先處理單位。真正重要的是、基本真理句子與預測句子之間的情緒差異。這可做為字錯誤率（WER）的Proxy。語音對文字的準確度很重要、但WER並未直接用於最終的傳輸途徑指標。

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

這些情緒指標可針對每個句子的F1分數、回收和精準度進行計算。然後可以將結果彙總並顯示在混淆對照表中、以及每個指標的可信度間隔。

使用轉移學習的好處在於、模型效能的提升只需一小部分的資料需求、訓練時間和成本。此外、還應將微調模型與基礎版本進行比較、以確保傳輸學習能提升效能、而非損害效能。換句話說、微調模式應該比預先訓練的模式更能在支援中心資料上執行。

管道評估

測試案例	詳細資料
測試編號	管道觀感指標
測試先決條件	針對語音對文字和情緒分析模式進行微調
預期結果	微調模型的觀感指標比原始的預先訓練模型更好。

管道觀感指標

- 1. 計算基礎模式的感受指標。
- 2. 計算微調模式的感受指標。
- 3. 計算這些指標之間的差異。
- 4. 平均所有句子的差異。

影片與示範

有兩部筆記型電腦包含意見分析管道： "「[Support-Center-Model-Transfer-Learning and 精細調校.ipynb](#)」" 和 "「[支援中心-意見分析- Pipele.ipynb](#)」"。這些筆記型電腦一起示範如何利用最先進的深度學習模式、針對使用者的資料進行微調、來開發管道來擷取支援中心資料、並從每一句話中擷取意見。

支援中心：情緒分析Pipeline.ipynb

這份筆記型電腦內含用於擷取音訊、將音訊轉換成文字、以及擷取用於外部儀表板的想法的Riva預估管道。如果尚未下載及處理資料集、則會自動下載及處理。筆記型電腦的第一部分是「語音對文字」、負責將音訊檔案轉換成文字。接下來是「意見分析」區段、該區段會擷取每個文字句子的意見、並以類似建議儀表板的格式顯示這些結果。



此筆記型電腦必須在模型訓練和微調之前執行、因為必須下載並轉換成正確格式的mp3資料集。

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

支援中心：模型訓練與微調.ipynb

在執行筆記型電腦之前、必須先設定TAO Toolkit虛擬環境（請參閱命令總覽中的TAO Toolkit一節以取得安裝說明）。

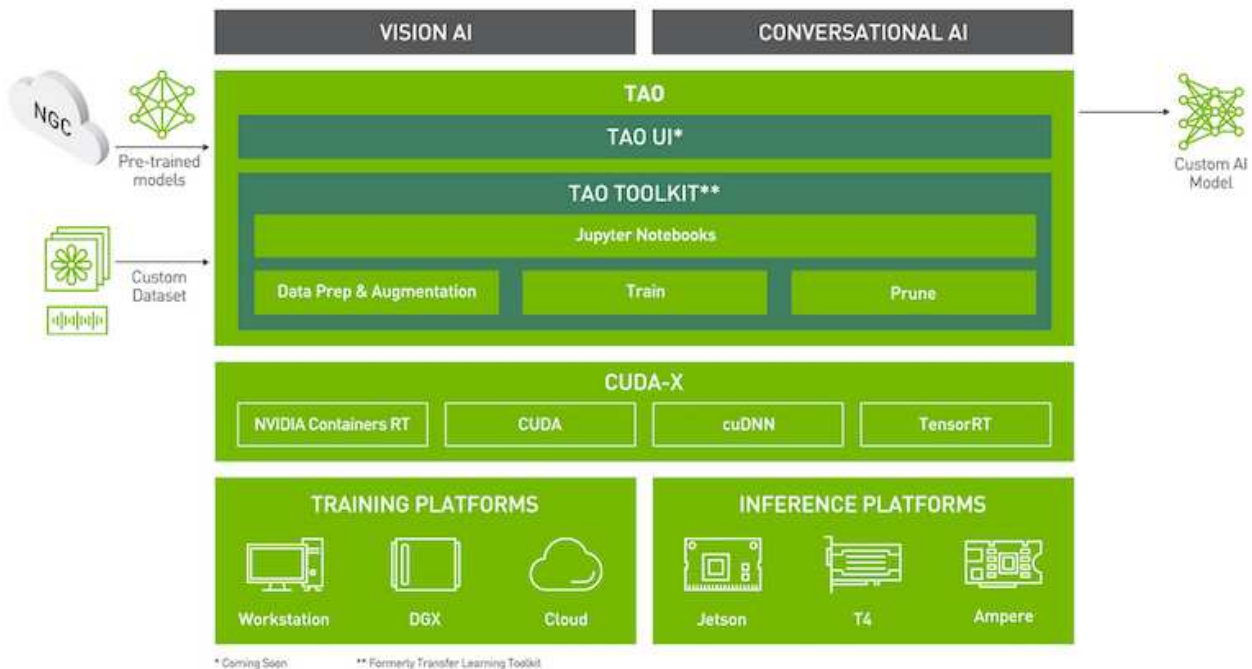
這款筆記型電腦仰賴TAO工具套件來微調客戶資料的深度學習模式。如同先前的筆記型電腦、此筆記分為兩個區段、分別用於語音對文字和情緒分析元件。每個區段都會進行資料處理、模型訓練和微調、結果評估和模型匯出。最後、我們會提供一個端點來部署您專為Riva所設計的兩種調校模型。

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.

結論

隨著客戶體驗逐漸被視為關鍵競爭戰場、人工智慧擴充的全球支援中心已成為幾乎每個產業的公司都無法忽視的關鍵元件。本技術報告所建議的解決方案已證實可支援提供如此優異的客戶體驗、現在的挑戰是確保企業採取行動、將AI基礎架構和 workflows 現代化。

AI在客戶服務中的最佳實作方式、不是取代人工代理程式。相反地、AI可透過即時情緒分析、爭議升級及多式多模式關聯運算、偵測口頭、非口頭和面向等暗示、讓他們有能力創造出卓越的客戶體驗、讓全方位的AI模式能夠大規模提出建議、並補充個別的人力代理商可能缺乏的內容。AI也能讓特定客戶與目前可用的代理程式更符合需求。企業可以利用AI、從他們對供應商產品、服務和品牌形象的想法和印象中、汲取寶貴的客戶意見。

此解決方案也可用來建構支援代理程式的時間序列資料、做為客觀的效能評估指標。傳統的客戶滿意度調查通常缺乏足夠的回應。藉由收集長期的員工與客戶意見、雇主可以根據充分資訊、針對支援代理商的績效做出決策。

NetApp、SFL Scientific、開放原始碼協調架構與NVIDIA的結合、將最新技術整合為託管服務、提供絕佳的靈活度、可加速技術採用、並縮短新的AI/ML應用程式的上市時間。這些進階服務是在內部部署、可輕鬆移轉至雲端原生環境及混合式部署架構。

何處可找到其他資訊

若要深入瞭解本文所述資訊、請檢閱下列文件和 / 或網站：

- 3D互動示範

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

- 直接與NetApp AI專家聯絡

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

- 採用NetApp解決方案簡介的NVIDIA基礎命令平台

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

- 適用於AI的NetApp 10大理由資訊圖表

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

- 醫療業AI：深入學習、識別肺CT掃描中的COVID-19病發白皮書

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

- 醫療業AI：監控醫療照護設定中的面板使用情況白皮書

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

- 醫療業AI：診斷影像技術報告

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

- AI for零售業：NetApp對話AI使用NVIDIA Riva

["https://docs.netapp.com/us-en/netapp-solutions/ai/cainvidia_executive_summary.html"](https://docs.netapp.com/us-en/netapp-solutions/ai/cainvidia_executive_summary.html)

- NetApp ONTAP 解決方案簡介

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

- NetApp DataOps Toolkit解決方案簡介

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

- NetApp AI Control Plane解決方案簡介

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

- 利用資料磁碟AI電子書改變產業

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

- NetApp EF系列AI解決方案簡介

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

- 適用於AI推斷解決方案簡介的NetApp AI和Lenovo ThinkSystem

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

- 適用於企業AI和ML解決方案簡介的NetApp AI和Lenovo ThinkSystem

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

- NetApp與NVIDIA–重新定義AI影片的可能性

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

Azure的分散式訓練：點選率預測

TR-4904：Azure中的分散式訓練-點擊率預測

Rick Huang、Verron Martina、Muneer Ahmad、NetApp

資料科學家的工作重點應放在機器學習（ML）和人工智慧（AI）模式的訓練和調校。不過、根據Google的研究結果、資料科學家約有80%的時間花在研究如何讓他們的模型與企業應用程式搭配運作、以及大規模執行。

若要管理端點對端點的AI/ML專案、需要更深入瞭解企業元件。雖然DevOps已接管定義、整合及部署、但這些類型的元件、但ML作業的目標是類似流程、包括AI/ML專案。若要瞭解企業中端點對端點的AI/ML傳輸途徑、請參閱下列必要元件清單：

- 儲存設備
- 網路
- 資料庫
- 檔案系統
- 容器
- 持續整合與持續部署（CI/CD）管道
- 整合式開發環境（IDE）
- 安全性
- 資料存取原則
- 硬體
- 雲端

- 虛擬化
- 資料科學工具集與程式庫

目標對象

資料科學領域涉及IT與企業的多個領域：

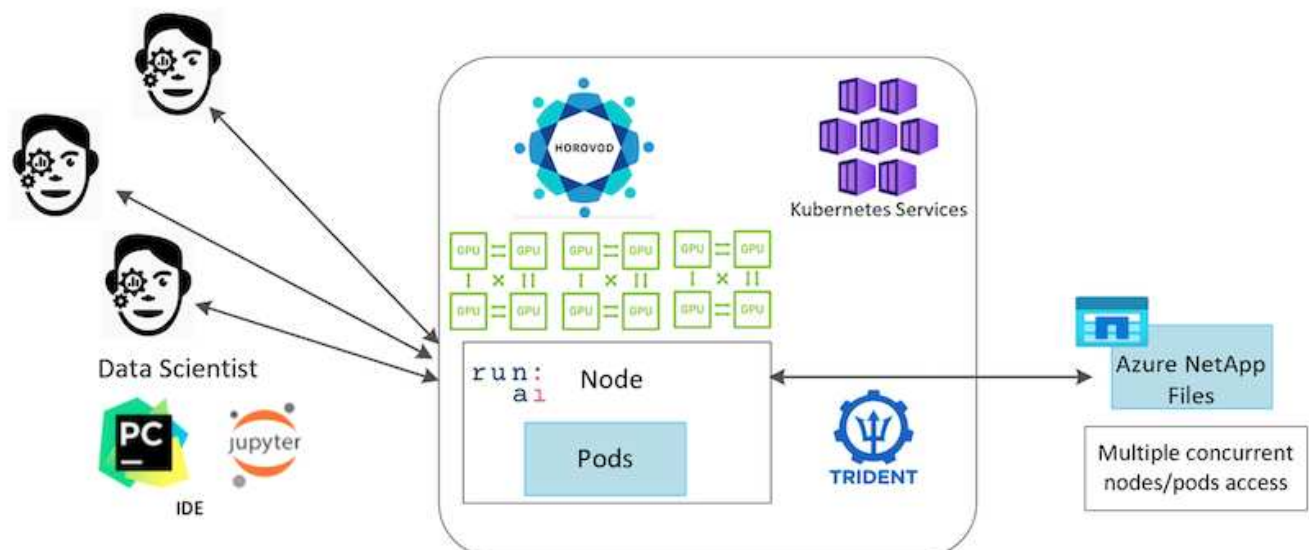
- 資料科學家需要靈活運用所選的工具和程式庫。
- 資料工程師需要知道資料的流通方式及存放位置。
- DevOps工程師需要工具、將新的AI/ML應用程式整合至其CI/CD管道。
- 雲端管理員與架構設計師必須能夠設定及管理Azure資源。
- 企業使用者想要存取AI / ML應用程式。

在本技術報告中、我們將說明Azure NetApp Files 各種角色的功能、包括功能性、速度性AI、dask和Azure如何協助企業實現價值。

解決方案總覽

此解決方案遵循AI / ML應用程式的生命週期。我們從資料科學家的工作開始、定義準備資料和訓練模型所需的不同步驟。我們運用dask上的快速發展、在Azure Kubernetes Service (aks) 叢集上執行分散式訓練、大幅縮短訓練時間、相較於傳統的Python scikr-k套 件學習方法。為了完成完整的週期、我們整合了整個管線Azure NetApp Files 與功能的功能。

提供多種效能等級。Azure NetApp Files客戶可以從標準層開始、在不中斷營運的情況下橫向擴充並擴充至高效能層、而不需移動任何資料。此功能可讓資料科學家在不發生任何效能問題的情況下、大規模訓練模型、避免跨叢集的任何資料封閉環境、如下圖所示。



技術總覽

本頁概述本解決方案所使用的技術。

Microsoft與NetApp

自2019年5月起、Microsoft針對企業NFS和SMB檔案服務提供Azure原生的第一方入口網站服務、以NetApp ONTAP 的更新技術為基礎。這項開發是由Microsoft與NetApp之間的策略合作關係所推動、進一步將世界級ONTAP 的流向資料服務延伸至Azure。

Azure NetApp Files

此功能是一項企業級的高效能計量檔案儲存服務。Azure NetApp Files支援任何工作負載類型、預設為高可用度。Azure NetApp Files您可以選擇服務和效能層級、並透過服務設定Snapshot複本。NetApp是Azure第一方服務、可在雲端移轉及執行最嚴苛的企業檔案工作負載、包括資料庫、SAP及高效能運算應用程式、而且不需變更程式碼。Azure NetApp Files

此參考架構可為IT組織提供下列優勢：

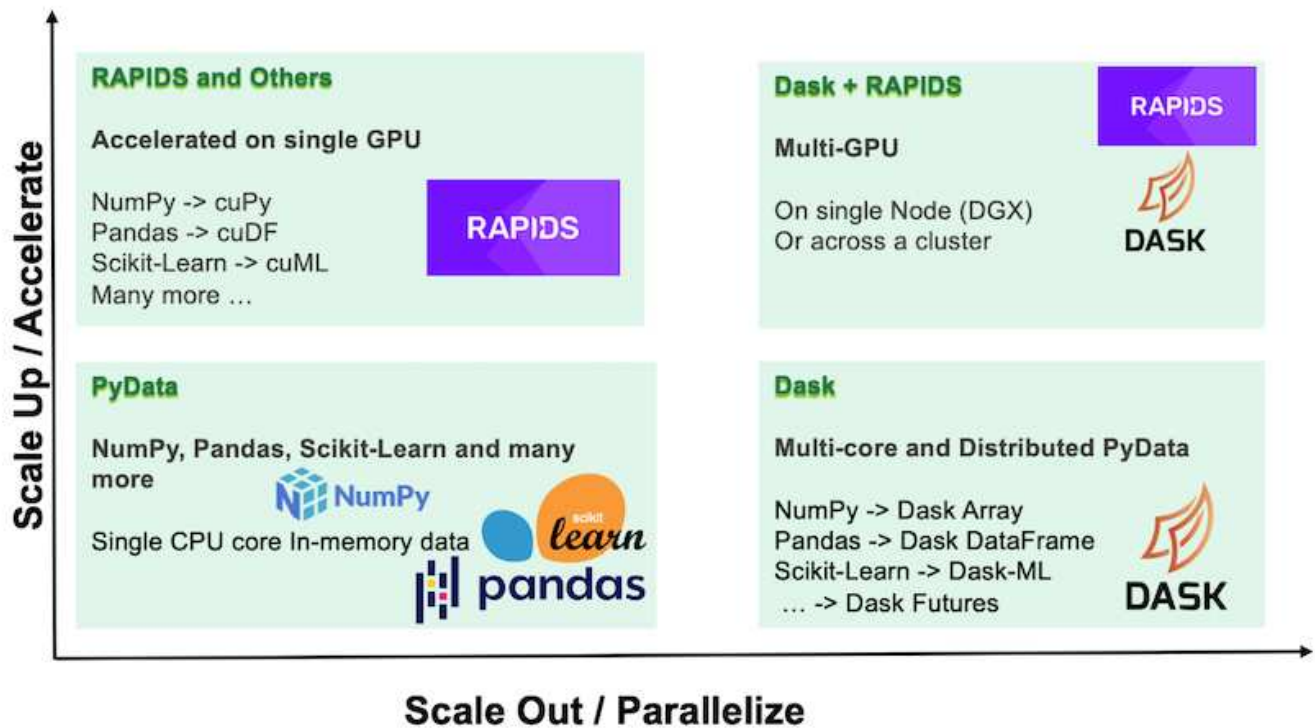
- 消除設計複雜性
- 可獨立擴充運算與儲存設備
- 讓客戶從小規模開始、並無縫擴充
- 提供多種儲存層、以滿足各種效能與成本效益

Dask與NVIDIA的水流總覽

dask是一款開放原始碼的平行運算工具、可在多部機器上擴充Python程式庫、並提供更快的大量資料處理速度。它提供的API類似於單執行緒傳統Python程式庫、例如pandas、Numbery和sciKit學習。因此、原生Python使用者不會被迫在現有程式碼中大幅變更、以使用整個叢集的資源。

NVIDIA Nirps是一套開放原始碼程式庫、可在GPU上執行端點對端點ML和資料分析工作流程。搭配dask、您可以輕鬆地從GPU工作站（垂直擴充）擴充至多節點、多GPU叢集（橫向擴充）。

若要在叢集上部署dask、您可以使用Kubernetes進行資源協調。您也可以根據程序需求來垂直擴充或向下擴充工作節點、如此有助於最佳化叢集資源使用量、如下圖所示。



軟體需求

下表列出此解決方案所需的軟體需求。

軟體	版本
Azure Kubernetes服務	1.18.14
水流和dask容器影像	儲存庫：「rapidsai/rapidsai」 標記：0.17-sula11.0-rime-ubuntu18.04
NetApp Trident	20.01.1
掌舵	3.0.00.0

雲端資源需求

本頁說明如何組態雲端資源Azure NetApp Files 以利執行功能。

設定Azure NetApp Files 功能

如Azure NetApp Files 所述設定功能 ["快速入門：設定Azure NetApp Files 功能以建立NFS磁碟區"](#)。

您可以跳過「Create NFS volume for Azure NetApp Files the Sfor the Sfor the」一節、因為您要透過Trident建立Volume。在繼續之前、請先完成下列步驟：

1. 註冊Azure NetApp Files 以取得NetApp資源供應商（透過Azure Shell）（["連結"](#)）。
2. 在Azure NetApp Files 以下網頁建立帳戶 ["連結"](#)）。
3. 設定容量資源池（最低4TB標準或Premium、視您的需求而定）（["連結"](#)）。下表列出在雲端設定的網路組態需求。dask叢集與Azure NetApp Files VMware必須位於同一個Azure虛擬網路（vnet）或對等虛擬網路

中。

資源	類型/版本
Azure Kubernetes服務	1.18.14
代理節點	3倍Standard_DS2_v2
GPU節點	3倍Standard_NC6s_v3
Azure NetApp Files	標準容量資源池
容量（以TB為單位）	4.

點選率預測使用案例摘要

此使用案例以公開提供的為基礎 "TB點選記錄" 資料集來源 "Criteo AI實驗室"。隨著ML平台與應用程式的最新進展、現在許多人都開始大規模學習。點選率（CTR）定義為每百個線上廣告曝光點選率（以百分比表示）的平均點選率。在各種產業垂直市場和使用案例中、包括數位行銷、零售、電子商務和服務供應商、廣泛採用此技術作為主要指標。以下是使用CTR作為潛在客戶流量重要指標的範例：

- 數位行銷： "Google Analytics"、CTR可用來衡量廣告商或商家的關鍵字、廣告及免費上市表現。高點選率是使用者發現您的廣告和清單有用且相關的良好指標。此外、CTR也有助於達成關鍵字的預期CTR、這是一項要素 "廣告排名"。
- *電子商務：*除了善用資源之外 "Google Analytics"中、電子商務後端至少有一些訪客統計資料。雖然這些統計資料乍看之下似乎並不實用、但通常很容易閱讀、而且可能比其他資訊更準確。由這類統計資料組成的第一方資料集為專屬資料、因此與電子商務賣家、買家及平台最相關。這些資料集可用來設定基準、比較去年和昨天的結果、建立時間序列以供進一步分析。
- *零售：*零售商店可將訪客數量和客戶數量與CTR建立關聯。從銷售點記錄可以看出客戶的數量。零售商網站的點選率或廣告流量可能導致上述銷售。忠誠度方案是另一個使用案例、因為客戶從線上廣告或其他網站重新導向、可能會加入以獲得獎勵。零售商可以透過忠誠度方案來贏得客戶、並記錄銷售記錄的行為、藉此建立一套建議系統、不僅能預測不同類別的消費者購買行為、也能將優惠券個人化、減少流失。
- *服務供應商：*電信公司和網際網路服務供應商擁有豐富的第一方使用者遙測資料、可提供深入見解的AI、ML和分析使用案例。舉例來說、電信公司可利用行動用戶的網路瀏覽、每日記錄最上層網域記錄、微調現有的模型、以製作最新的對象區隔、預測客戶行為、並與廣告商合作、以即時廣告方式提供更好的線上體驗。在這類資料導向的行銷工作流程中、CTR是反映轉換的重要指標。

就數位行銷而言、"請按一下「Logs（記錄）」" 現在是評估ML平台與演算法擴充性的參考資料集。透過預測點選率、廣告商可以選擇最有可能回應廣告的訪客、分析他們的瀏覽記錄、並根據使用者的興趣顯示最相關的廣告。

本技術報告所提供的解決方案強調下列優點：

- 分散式或大規模訓練的優勢Azure NetApp Files
- 支援CUDA的資料處理（cuDF、cuPy等）和ML演算法（cuML）
- 適用於分散式訓練的dask平行運算架構

端點對端點工作流程建立在「水流AI」和Azure NetApp Files「速度AI」基礎上、顯示隨機樹系模式訓練時間大幅縮短兩個訂單。相較於傳統的「大作」方法、這項改善相當顯著、因為在處理實際的點擊記錄時、每天都會有45GB的結構化表格資料（平均）。這相當於包含約20億列的DataFrame。我們將示範叢集環境設定、架構與

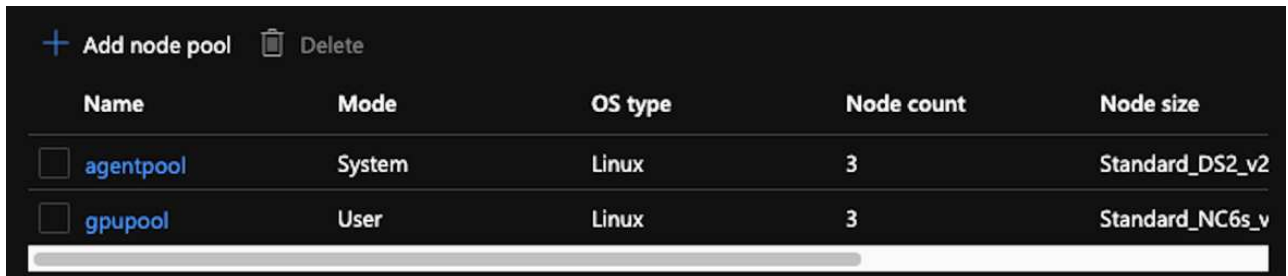
程式庫安裝、資料載入與處理、傳統與分散式訓練、視覺化與監控、並在本技術報告中比較關鍵的端點對端點執行時間結果。

設定

安裝及設定高效能叢集

若要安裝及設定AKS叢集、請參閱網頁 ["建立一個高效能叢集"](#) 然後完成下列步驟：

1. 選取節點類型（系統[CPU]或工作者[GPU]節點）時、請選取下列項目：
 - a. 主系統節點應為標準DS2v2（「agentpool」預設三個節點）。
 - b. 然後為名為「gp池」的使用者群組（GPU節點）新增工作節點Standard_NC6s_v3集區（最少三個節點）。



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分鐘。完成後、按一下「Connect to Cluster（連線至叢集）」。
3. 若要連線至新建立的高層叢集、請從您的本機環境（筆記型電腦/電腦）安裝下列項目：
 - a. Kubernetes命令列工具使用 ["針對您的特定作業系統所提供的說明"](#)
 - b. Azure CLI、如文件所述、["安裝Azure CLI"](#)
4. 若要從終端機存取高層叢集、請輸入「AZ登入」並輸入認證資料。
5. 執行下列兩個命令：

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

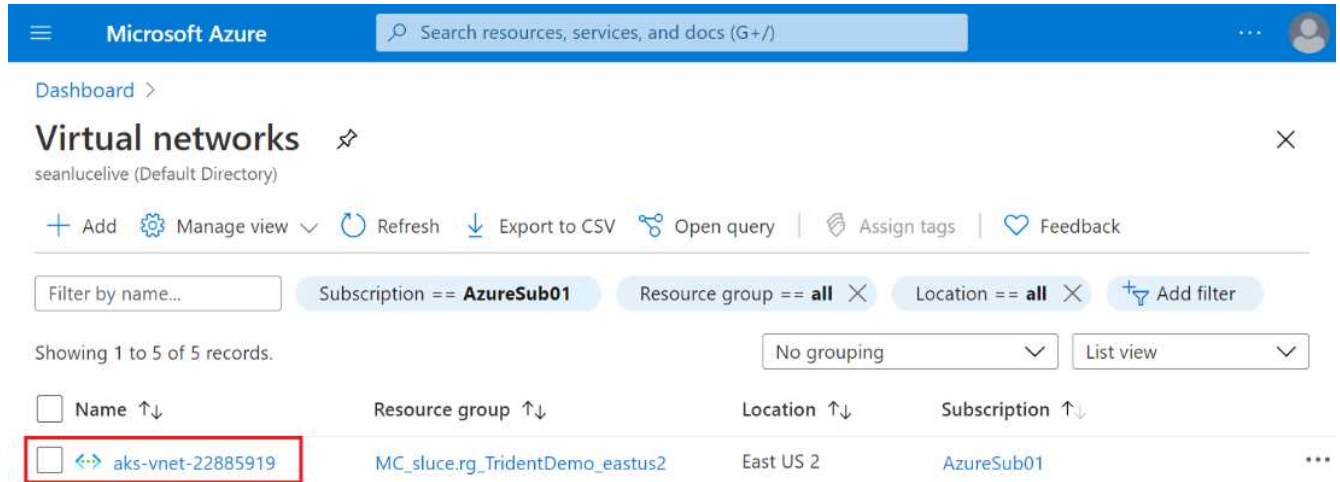
6. 輸入「Azure CLI：kubectl Get nodes」。
7. 如果所有六個節點都已啟動並執行、如下列範例所示、則您的高效能叢集已就緒並連線至本機環境

```
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 ~ %
```

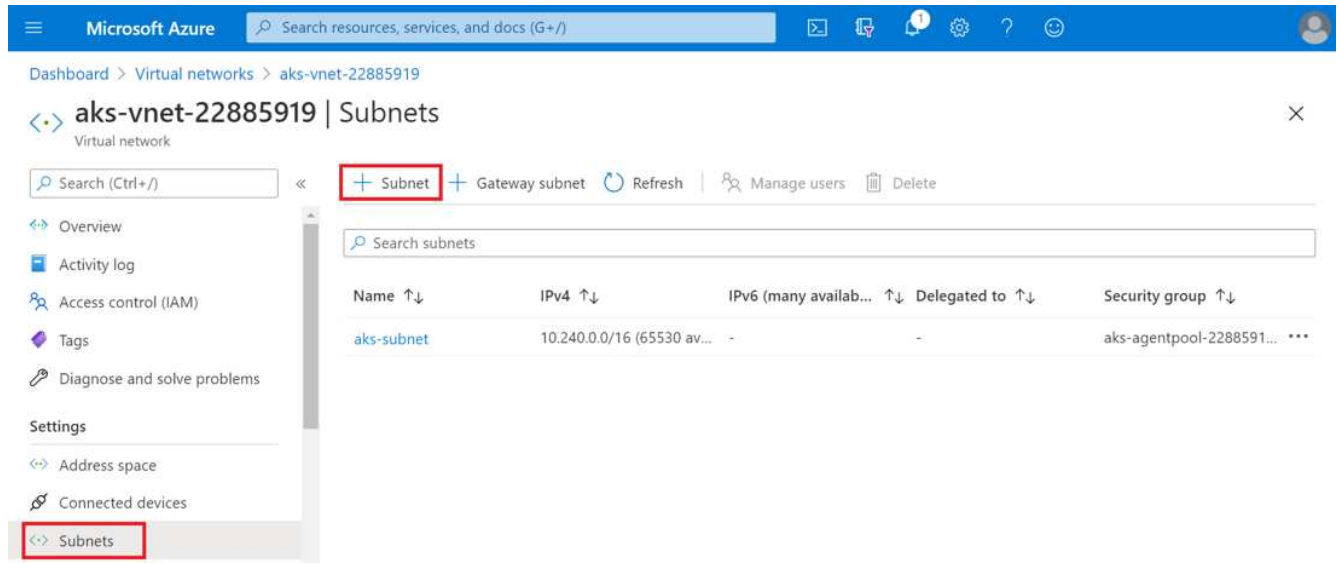
建立委派子網路Azure NetApp Files 以供使用

若要建立Azure NetApp Files 委派的子網路以供使用、請完成下列步驟：

1. 瀏覽至Azure入口網站內的虛擬網路。尋找您新建立的虛擬網路。它應該有一個前置詞、例如「pak vnet」。
2. 按一下vnet的名稱。



3. 按一下子網路、然後按一下頂端工具列的+子網路。



4. 提供子網路名稱、例如「anf.sn」、然後在「Subnet委派」標題下選取「microsoft.Netapp/volumes」。請勿變更任何其他項目。按一下「確定」。

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

將實體磁碟區分配給應用程式叢集、並作為Kubernetes中的持續磁碟區宣告（PVCS）使用。Azure NetApp Files而這個程序也能讓您靈活地將它們對應到不同的服務、例如Jupyter筆記型電腦、無伺服器功能等。

服務使用者可從平台使用多種儲存設備。在本技術報告討論NFSs的同時、Azure NetApp Files 本產品的主要優點是：

- 讓使用者能夠使用Snapshot複本。
- 讓使用者能夠將大量資料儲存在Azure NetApp Files 功能區上。
- 在Azure NetApp Files 大量檔案上執行模型時、請善用效能優勢。

Peer aks vnet和Azure NetApp Files vnet

若要將網路上的高階客戶端與Azure NetApp Files the整套系統對等、請完成下列步驟：

1. 在搜尋欄位中輸入「Virtual Networks（虛擬網路）」。
2. 選取「vnet aks：vnet-name。」 按一下搜尋欄位、然後輸入產品。
3. 按一下「+新增」。
4. 輸入下列描述元：
 - a. 對等連結名稱為「aks-vnet-name_to_anf」。
 - b. SubscriptionID和Azure NetApp Files 以vnet對等合作夥伴身分執行的功能。
 - c. 保留所有非星號區段的預設值。
5. 按一下「新增」

如需詳細資訊、請參閱 "[建立、變更或刪除虛擬網路對等關係](#)"。

安裝Trident

若要使用Helm安裝Trident、請完成下列步驟：

1. 安裝Helm（如需安裝指示、請造訪 "[來源](#)"）。
2. 下載並解壓縮Trident 20.01.1安裝程式。

```
$wget  
$tar -xf trident-installer-21.01.1.tar.gz
```

3. 將目錄變更為「Trident安裝程式」。

```
$cd trident-installer
```

4. 將「tridentctl」複製到系統「\$path」中的目錄。

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

5. 在Kubernetes（K8s）叢集上使用Helm（"[來源](#)"）：

- a. 將目錄變更為「helm」目錄。

```
$cd helm
```

- b. 安裝Trident。

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

- c. 檢查Trident Pod的狀態。

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

如果所有的Pod都已啟動並正在執行、則會安裝Trident、您可以繼續前進。

6. 設定Azure NetApp Files 適用於高效能的不中斷端和儲存類別。

- a. 建立Azure服務原則。

服務宗旨是Trident如何與Azure溝通、以操控Azure NetApp Files 您的整套資源。

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

輸出應如下所示：

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

7. 建立Trident後端json檔案、例如名稱「anf-backend.json」。
8. 使用您偏好的文字編輯器、在「anf-backend.json」檔案中填寫下列欄位：

```
{
  "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. 請替換下列欄位：

- 《訂閱ID》。您的Azure訂閱ID。
- 「TenantId」。您的Azure租戶ID、來自上一步「AZ廣告服務」的輸出。
- "clientID"。您的應用程式ID來自前一步驟的「AZ廣告服務」輸出。
- 「客戶機密」。您的密碼來自上一步「AZ廣告服務」的輸出。

10. 指示Trident使用Azure NetApp Files 「anf-backend.json」作為組態檔、在「Trident」命名空間中建立支援功能的後端：

```
$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. 建立儲存類別。Kubernetes使用者透過PVCS來配置磁碟區、這些PVCS會依名稱指定儲存類別。指示K8s建立儲存類別「azurenetaappfiless」、以參照上一步建立的Trident後端。
12. 建立Yaml（「anf-storage - class.yaml」）檔案、以供儲存類別和複製。


```

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. 驗證是否已建立儲存類別。

```
kubectl get sc azurenetappfiles
```

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

使用Helm在高峰上設定dask、並部署快速部署

若要使用Helm在高峰上設定具有快速部署功能的dask、請完成下列步驟：

1. 建立命名空間、以安裝具有快速功能的dask。

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

2. 建立一個PVC來儲存點選率資料集：

a. 將下列Yaml檔案內容儲存至檔案以建立永久保存。

```

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

```

b. 將Yaml檔案套用至Kubernetes叢集。

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

3. 複製「rapidsai git」儲存庫（"<https://github.com/rapidsai/helm-chart>"）。

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

4. 修改「values.yaml」、並加入先前為工作者和Jupyter工作區所建立的永久虛擬網路。

- a. 轉到儲存庫的「rapidsai」目錄。

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

- b. 更新「values.yaml」檔案、然後使用PVC掛載磁碟區。

```
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. 移至儲存庫的主目錄、然後使用Helm在使用者節點上部署具有三個工作節點的dask。

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

效能等級 Azure NetApp Files

您可以將磁碟區移至另一個容量集區、使用您要的磁碟區服務層級、藉此變更現有磁碟區的服務層級。此解決方案可讓客戶從標準層級的小型資料集和少量GPU開始著手、並隨著資料量和GPU增加而橫向擴充或垂直擴充至優質層級。頂級層提供的處理量是標準層級的四倍、而垂直擴充則無需移動任何資料即可變更磁碟區的服務層級。

動態變更磁碟區的服務層級

若要動態變更磁碟區的服務層級、請完成下列步驟：

1. 在「Volumes（磁碟區）」頁面上、以滑鼠右鍵按一下您要變更其服務層級的磁碟區。選取變更資源池。

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「變更資源池」視窗中、選取您要將磁碟區移至的容量資源池。



3. 按一下「確定」。

自動化效能層級變更

下列選項可用於自動化效能層級變更：

- 動態服務層級變更目前仍在「公開預覽」中、預設不會啟用。若要在Azure訂閱上啟用此功能、請參閱本說明文件、瞭解如何使用 ["動態變更磁碟區的服務層級"](#)。
- 中提供Azure CLI Volume Pool變更命令 ["Volume Pool變更文件"](#) 以及下列範例：

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

- PowerShell ["Set-AzNetAppFilesVolume Pool Cmdlet"](#) 變更Azure NetApp Files 一個聲音區的資源池、如下例所示：

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

點選率預測資料處理與模型訓練

用於資料處理和模型訓練的程式庫

下表列出用來建置此工作的程式庫和架構。所有這些元件均已與Azure的角色型存取與安全控管功能完全整合。

程式庫/架構	說明
dask cuML	若要讓ML在GPU上運作、請使用 "CUML程式庫" 提供使用dask存取水流立方ML套件的功能。藉由以GPU為基礎的高效能實作、讓您在處理CPU的基礎上、能夠以高達100倍的速度加速、藉由使用NetApp的功能來實作熱門的ML演算法、包括叢集、維度減量及回歸方法。
dask cudf	CUDF包含多種其他功能、可支援GPU加速擷取、轉換、負載（ETL）、例如資料子設定、轉換、單一熱編碼等。水流團隊維持 "dAsk擁抱程式庫" 這包括使用dask和cuDF的輔助程式方法。
科學套件學習	SciPKit學習提供數十種內建的機器學習演算法和模型、稱為評估工具。每個 "預估工具" 可搭配使用的部分資料 "適合" 方法。

我們使用兩部筆記型電腦來建構ML管線進行比較、其中一部是傳統的「大大管」學習方法、另一部則是以「急水流」和「dask」進行分散式訓練。每一部筆記型電腦均可個別測試、以瞭解其在時間與擴充方面的效能表現。我們會個別介紹每一部筆記型電腦、以展示使用「水流」和「dask」進行分散式訓練的優點。

Load Criteo按一下「日誌第15天」（日誌第15天）、然後訓練sci套 件學習隨機樹系模式

本節說明我們如何使用Pandas和dask DataFrames從Criteo Terabyte資料集載入Click記錄資料。使用案例與數位廣告有關、可預測是否要點選廣告、或是Exchange未在自動化管道中使用準確的模式、藉此建立使用者的設定檔。

我們從Click Logs資料集載入第15天的資料、總計45GB。在Jupyter筆記型電腦「CTer-andasRF-collated.ipynb」中執行下列儲存格、會建立一個包含前5、000萬列的大圓子資料框架、並產生scipit-記憶 隨機樹系模型。

```

%%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)

```

若要使用受過訓練的隨機樹系模型來執行預測、請執行本筆記型電腦的下一段。我們從第15天開始採用最後一百萬列作為測試集、以避免任何重複資料。儲存格也會計算預測的準確度、定義為模型準確預測使用者是否點選廣告的發生百分比。若要檢閱此筆記型電腦中任何不熟悉的元件、請參閱 ["正式的sci套件學習文件"](#)。

```
# 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**中載入第15天、並訓練**dask cuML**隨機樹系模型

以類似上一節的方式、在Pandas中載入Criteo按一下「日誌第15天」、然後訓練scier-記憶隨機樹系模式。在此範例中、我們使用dask couDF執行DataFrame載入、並在dask cuML中訓練隨機樹系模型。我們比較了本節訓練時間與規模的差異 "[「訓練時間比較」](#)。"

criteo_dASK_RF.ipynb

本筆記型電腦會匯入「umpy」、「累計」及必要的「dask」程式庫、如下列範例所示：

```
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
```

啟動dask Client()。

```
client = Client()
```

如果您的叢集設定正確、您可以看到工作節點的狀態。


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

在我們的高層叢集中、會顯示下列狀態：

Client	Cluster
Scheduler: tcp://rapidsai-scheduler:8786	Workers: 3
Dashboard: /proxy/rapidsai-scheduler:8787/status	Cores: 3
	Memory: 354.55 GB

請注意、dask採用的是閒置執行模式：dask並不是立即執行處理程式碼、而是建立直接執行的Acyclic圖表（DAG）。DAG包含一組工作及其互動、每位員工都需要執行這些工作。此配置表示在使用者指示dask以某種方式執行工作之前、工作不會執行。有了dask、您有三個主要選項：

- *在DataFrame上呼叫compact()。*此呼叫會處理所有分割區、然後將結果傳回排程器、以供最終集合併轉換至cuDF DataFrame。除非排程器節點的記憶體不足、否則此選項應謹慎使用、且僅用於大幅減少的結果。
- * DataFrame上的呼叫持續 ()。*此呼叫會執行圖表、但它不會將結果傳回排程器節點、而是將結果保留在整個叢集的記憶體中、讓使用者無需重新執行相同的處理、即可在管線中重複使用這些中繼結果。
- * DataFrame上的呼叫標頭 ()。*就像cuDF一樣、此呼叫會傳回10筆記錄給排程器節點。此選項可用來快速檢查DataFrame是否包含所需的輸出格式、或記錄本身是否合理、視處理和計算而定。

因此、除非使用者撥打上述任一動作、否則工作人員會閒置等待排程器啟動處理。這種閒置執行模式在Apache Spark等現代化平行與分散式運算架構中相當常見。

以下段落使用dask cuML來訓練隨機樹系模型、以進行分散式GPU加速運算、並計算模型預測準確度。

```

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 )

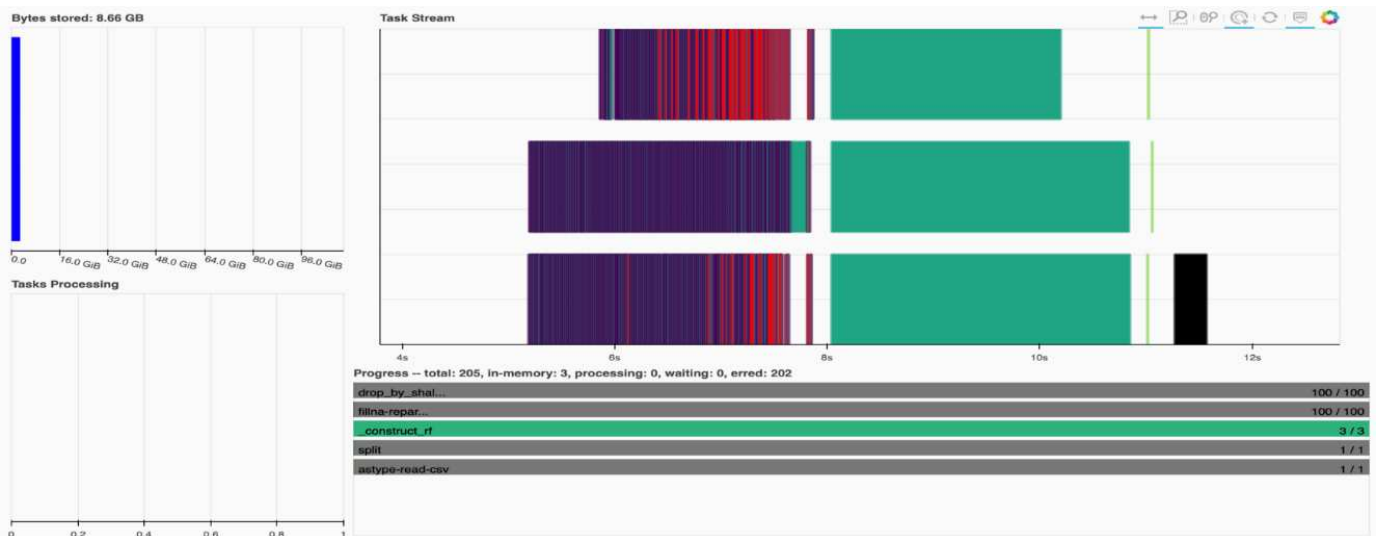
```

使用原生工作串流儀表板監控**dsask**

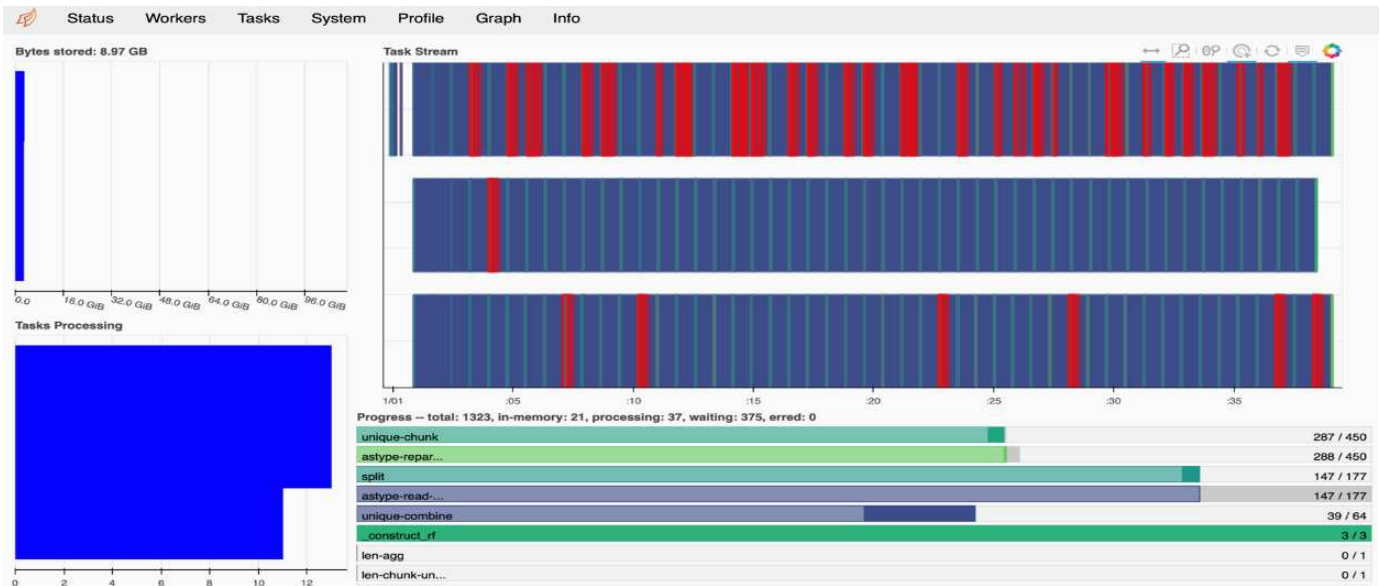
◦ **"dask分散式排程器"** 提供兩種形式的即時回饋：

- 互動式儀表板包含許多繪圖和含有即時資訊的表格
- 進度列適合在主控台或筆記型電腦中互動使用

在我們的案例中、下圖顯示如何監控工作進度、包括儲存的位元組、詳細細分串流數量的工作串流、以及執行相關功能的工作名稱進度。在我們的案例中、因為我們有三個工作節點、所以串流有三個主要區塊、而且色彩代碼會在每個串流中指出不同的工作。



您可以選擇分析個別工作、以毫秒為單位檢查執行時間、或找出任何障礙或阻礙。例如、下圖顯示隨機樹系模型擬合階段的工作串流。執行的功能相當多、包括用於DataFrame處理的獨特區塊、用於調整隨機樹系的_Constrature_RF等。大多數時間都花在DataFrame作業上、因為Criteo Click記錄中一天資料的大小（45GB）太大。



訓練時間比較

本節比較使用傳統Pandas的模型訓練時間與dask。對於Pandas而言、由於處理時間變慢、因此載入的資料量較少、以避免記憶體溢位。因此、我們將結果插補以提供公平的比較。

下表顯示隨機樹系模型使用的資料大幅減少時的原始訓練時間比較（資料集的20億個資料集中、有500萬列的資料）。此範例僅使用所有可用資料的0.25%以下。而對於dASK CUMML、我們則針對所有200億個可用資料列、訓練隨機樹系模型。這兩種方法可提供相當的訓練時間。

方法	訓練時間
科學套件學習：在第15天只使用50M列做為訓練資料	47分21秒
「激流勇進」：將第15天的全部20B列當作訓練資料	1小時12分鐘11秒

如果我們以線性方式插補訓練時間結果、如下表所示、則搭配使用dask的分散式訓練將有顯著的優勢。傳統的「大作大作」學習方法需要13天的時間來處理和訓練45GB的資料、只需一天的點擊記錄、而「大浪」方法則能以相同數量的資料處理速度快262.39倍。

方法	訓練時間
科學套件-學習：將第15天的所有20B列當作訓練資料	13天、3小時、40分鐘及11秒
「激流勇進」：將第15天的全部20B列當作訓練資料	1小時12分鐘11秒

在上表中、您可以看到、透過使用配備DASK的PRUs將資料處理和模型訓練分散到多個GPU執行個體、相較於使用sciker-Learn模型訓練的傳統Pandas DataFrame處理、執行時間大幅縮短。此架構可在多節點、多GPU叢集內、在雲端及內部部署中進行橫向擴充。

利用Prometheus和Grafana監控dask和水流

部署所有項目之後、請針對新資料執行推斷。這些模型會根據瀏覽活動來預測使用者是否點選廣告。預測結果會儲存在dask couDF中。您可以使用Prometheus監控結果、並在Grafana儀表板中以視覺化的方式呈現結果。

如需詳細資訊、請參閱 ["中速漂流"](#)。

使用NetApp DataOps Toolkit的資料集與模型版本管理

NetApp DataOps Toolkit for Kubernetes可將儲存資源與Kubernetes工作負載抽象化、直到資料科學工作區層級。這些功能均以簡單易用的介面進行封裝、專為資料科學家和資料工程師所設計。使用熟悉的Python程式形式、工具套件可讓資料科學家和工程師在短短數秒內配置及銷毀JupyterLab工作區。這些工作區可包含TB甚至PB的儲存容量、讓資料科學家能夠將所有訓練資料集直接儲存在專案工作區中。現在已經不再需要分別管理工作區和資料磁碟區了。

如需詳細資訊、請參閱工具套件 ["GitHub儲存庫"](#)。

Jupyter筆記型電腦供參考

本技術報告有兩部Jupyter筆記型電腦：

- ["* CTR - pastasRF-collated.ipynb.*"](#) 這款筆記型電腦會從Criteo TB Click日誌資料集載入第15天的資料、將資料處理及格式化為子網頁資料框架、訓練科學套件學習隨機樹系模型、執行預測並計算準確度。
- ["* criteo_dAsk_RF.ipynb.*"](#) 本筆記型電腦會從Criteo TB載入第15天的內容、按一下「記錄資料集」、將資料處理及格式化為dask couDF、訓練dask cuML隨機樹系模型、執行預測並計算準確度。藉由運用GPU來運用多個工作節點、這種分散式資料與模型處理與訓練方法效率極高。您處理的資料越多、相較於傳統的ML方法、所節省的時間就越多。您可以將此筆記型電腦部署在雲端、內部部署或混合式環境、其中Kubernetes叢集包含不同位置的運算和儲存設備、只要您的網路設定能夠自由移動資料和模型發佈。

結論

藉由整合Docker和Kubernetes等協調工具、可加速並簡化大規模ML處理與訓練的部署。Azure NetApp Files透過統一化端點對端點資料傳輸途徑、此解決方案可降低許多進階運算工作負載固有的延遲與複雜度、有效地彌補開發與營運之間的落差。資料科學家可以在大型資料集上執行查詢、並在訓練階段與其他使用者安全地共用資料和演算法模型。

在建置自己的AI / ML管道時、設定架構中元件的整合、管理、安全性及存取能力、是一項艱鉅的任務。讓開發人員存取及控制其環境是另一組挑戰。

透過在雲端建置端點對端點分散式訓練模式和資料傳輸途徑、相較於未採用GPU加速資料處理和運算架構的傳統開放原始碼方法、我們在工作流程完成時間總計方面、表現出兩大進步。

NetApp、Microsoft、開放原始碼協調架構和NVIDIA的結合、將最新的技術整合為託管服務、提供絕佳的靈活度、可加速技術採用、並縮短新的AI/ML應用程式的上市時間。這些進階服務是在雲端原生環境中提供、可輕鬆移轉至內部部署和混合式部署架構。

何處可找到其他資訊

若要深入瞭解本文所述資訊、請參閱下列資源：

- 產品特色：Azure NetApp Files
 - 解決方案架構頁Azure NetApp Files 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)

- Trident持續儲存容器：

- 可與Trident一起使用Azure NetApp Files

["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和水流：

- dask

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

- 安裝dsask

["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機器學習

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

- DAsk分散式診斷

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

- ML架構與工具：

- TensorFlow：適用於所有人的開放原始碼機器學習架構

["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/)

- Jupyter筆記型電腦伺服器

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

TR-4896：Azure的分散式訓練：線道偵測-解決方案設計

NetApp Ronnen Dar的Muneer Ahmad和Verron Martina、RUN：AI

自2019年5月起、Microsoft提供Azure原生的第一方入口網站服務、以NetApp ONTAP 的更新技術為基礎、提供企業NFS和SMB檔案服務。這項開發是由Microsoft與NetApp之間的策略合作關係所推動、進一步將世界級ONTAP 的流向資料服務延伸至Azure。

NetApp是領先業界的雲端資料服務供應商、已與RUN攜手合作：AI是一家虛擬化AI基礎架構的公司、能夠以完全GPU使用率來加速AI實驗。合作夥伴關係可讓團隊同時執行多項實驗、快速存取資料、並善用無限運算資源、進而加速AI運作。RUN：AI透過自動化資源配置來實現完整的GPU使用率、而獲證實的Azure NetApp Files 架構則可消除資料管線障礙、讓每個實驗都能以最快速度執行。

NetApp與RUN：AI聯手為客戶提供一個符合未來需求的平台、讓客戶在Azure中享有AI體驗。從分析、高效能運算（HPC）、到自主決策（客戶只需支付所需的費用、就能將IT投資最佳化）、NetApp與RUN之間的聯盟：AI在Azure Cloud提供單一的統一化體驗。

解決方案總覽

在此架構中、重點在於AI或機器學習（ML）分散式線道偵測訓練程序中運算密集程度最高的部分。車道偵測是自主駕駛中最重要的任務之一、可協助引導車輛定位車道標記。諸如車道標記等靜態元件可引導車輛以互動且安全的方式在高速公路上行駛。

複雜的神經網路（有線電視新聞網）型方法、已將場景的瞭解和分段推向新的境界。雖然對於可能會被遮蔽的長結構和區域（例如、柱狀物、線上陰影等）物件來說、它的效能並不理想。空間凸狀神經網路（SCNN）可將有線電視新聞網（CNN）泛化為豐富的空間層級。它可在同一層的新學之間傳播資訊、因此最適合用於通道、極或阻塞的卡車等結構化物件。這種相容性是因為空間資訊可以強化、而且可以保持順暢度和連續性。

需要在系統中注入數千個場景影像、以便讓模型學習並區分資料集中的各種元件。這些影像包括天氣、白天或夜間、多線道公路及其他交通路況。

在訓練方面、需要良好的資料品質和數量。單一GPU或多個GPU可能需要數天到數週的時間才能完成訓練。資料分散式訓練可以使用多個和多個GPU來加速程序。Horovod是這樣的架構、可提供分散式訓練、但讀取GPU叢集內的資料可能會成為障礙。提供超快、高處理量和持續低延遲、以提供橫向擴充/橫向擴充功能、讓GPU發揮最大運算能力。Azure NetApp Files我們的實驗證實、叢集內的所有GPU平均使用超過96%的時間來使用SCNN進行線道偵測訓練。

目標對象

資料科學在IT和業務中納入多項原則、因此我們的目標對象包括多位人員：

- 資料科學家需要靈活運用自己選擇的工具和程式庫。
- 資料工程師需要知道資料的流通方式及存放位置。
- 自主駕駛使用案例專家。
- 雲端管理員與架構設計師、負責設定及管理雲端（Azure）資源。
- DevOps工程師需要工具、將新的AI/ML應用程式整合到他們的持續整合與持續部署（CI/CD）管線中。
- 企業使用者想要存取AI / ML應用程式。

在本文件中、我們將說明Azure NetApp Files 功能完善、執行：AI和Microsoft Azure如何協助這些角色為企業帶來價值。

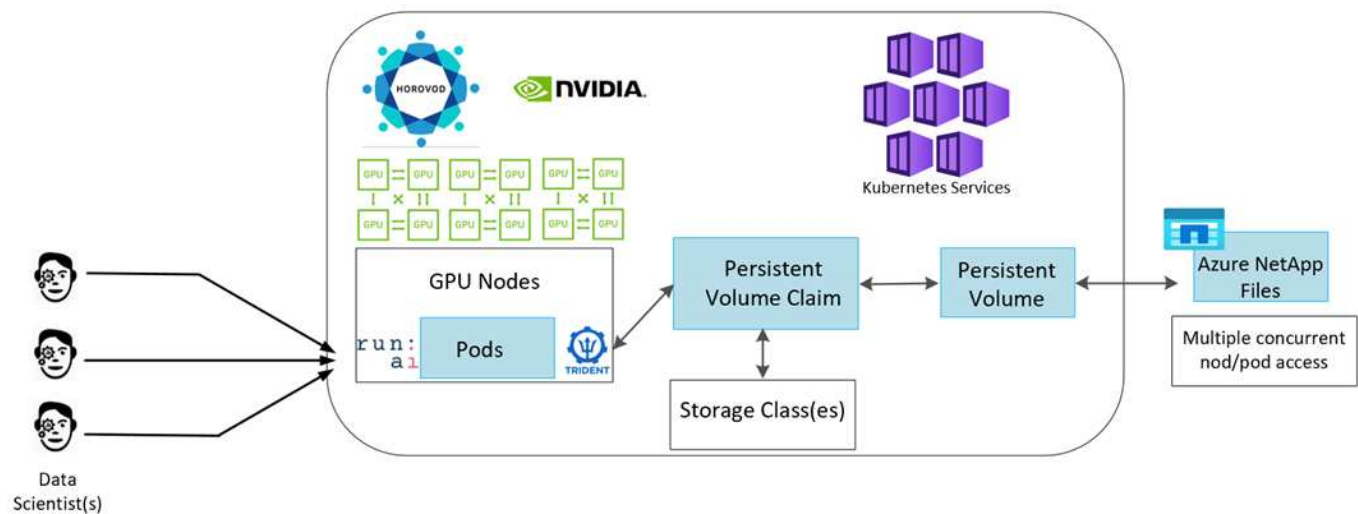
解決方案技術

本節說明線道偵測使用案例的技術需求、方法是大規模實作可在Azure雲端中完整執行的分散式訓練解決方案。下圖概述解決方案架構。

本解決方案所使用的元素包括：

- Azure Kubernetes服務（KS）
- Azure運算SKU搭配NVIDIA GPU
- Azure NetApp Files
- 執行：AI
- NetApp Trident

此處提及的所有元素連結均列於中 "其他資訊" 區段。



雲端資源與服務需求

下表列出實作解決方案所需的硬體元件。在任何解決方案實作中使用的雲端元件、可能會因客戶需求而異。

雲端	數量
高	至少三個系統節點和三個GPU工作節點
虛擬機器（VM）SKU系統節點	三個Standard_DS2_v2
VM SKU GPU工作節點	三個Standard_NC6s_v3
Azure NetApp Files	4TB標準層

軟體需求

下表列出實作解決方案所需的軟體元件。在解決方案的任何實作中使用的軟體元件、可能會因客戶需求而異。

軟體	版本或其他資訊
KS - Kubernetes版本	1.18.14

軟體	版本或其他資訊
執行：AI CLI	v2.2.25
RUN：AI Orchestration Kubernetes運算子版本	1.0.109.
霍羅沃德	0.21.2
NetApp Trident	20.01.1
掌舵	3.0.00.0

線道偵測—採用**RUN：AI**的分散式訓練

本節詳細說明如何使用**RUN：AI Orchestrator**設定平台、以大規模執行線上偵測分散式訓練。我們將討論如何在上述平台上安裝所有解決方案元素、以及執行分散式訓練工作。使用與**RUN：AI**實驗連結的**NetApp Snapshot™**來完成ML版本管理、以實現資料與模型的可重複性。ML版本管理在追蹤模型、團隊成員之間的共享工作、結果的重現性、將新的模型版本推向正式作業、以及資料來源等方面扮演著重要角色。**NetApp ML版本控制**

(Snapshot)可擷取與每個實驗相關的資料、訓練模型和記錄的時間點版本。它提供豐富的API支援、可輕鬆與**RUN：AI**平台整合、您只需根據訓練狀態觸發事件即可。您也必須擷取整個實驗的狀態、而不需變更程式碼或在Kubernetes (K8s) 上執行的容器中的任何內容。

最後、本技術報告會在各個高架（大量）的多個GPU節點上進行效能評估、以做為總結。

使用**TuSimple**資料集的線道偵測使用案例分散式訓練

在本技術報告中、我們會在**TuSimple**資料集上執行分散式訓練、以偵測線道。**Horovod**用於訓練程式碼、可在Kubernetes叢集中的多個GPU節點上同時透過高層管理程式進行資料分散式訓練。程式碼會封裝成容器映像、以便下載及處理**TuSimple**資料。已處理的資料儲存在**NetApp Trident**外掛程式所配置的持續磁碟區上。在訓練過程中、會再建立一個容器映像、並使用下載資料時所建立之持續磁碟區中儲存的資料。

若要提交資料和訓練工作、請使用**RUN：AI**來協調資源分配和管理。**RUN：AI**可讓您執行**Horovod**所需的訊息傳遞介面 (MPI) 作業。此配置可讓多個GPU節點彼此通訊、以便在每次訓練迷你批次後更新訓練權重。它也能透過UI和CLI監控訓練、讓您輕鬆監控實驗進度。

NetApp Snapshot已整合於訓練程式碼中、可擷取每個實驗的資料狀態和訓練模式。此功能可讓您追蹤所使用的資料和程式碼版本、以及所產生的相關訓練模型。

設定與安裝

如需設定及安裝AKS叢集、請前往 ["建立一個高效能叢集"](#)。接著、請遵循下列一系列步驟：

1. 選取節點類型（無論是系統（CPU）或工作（GPU）節點）時、請選取下列項目：
 - a. 在「tandard_DS2_v2」大小新增名為「agentpool」的主要系統節點。使用預設的三個節點。
 - b. 使用「Standard_NC6s_v3」資源池大小新增工作節點「gp池」。GPU節點至少使用三個節點。

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



部署需要5–10分鐘。

- 部署完成後、按一下「Connect to Cluster（連線至叢集）」。若要連線至新建立的高層叢集、請從本機環境（筆記型電腦/電腦）安裝Kubernetes命令列工具。請造訪 ["安裝工具"](#) 以根據您的作業系統進行安裝。
- ["在您的本機環境中安裝Azure CLI"](#)。
- 若要從終端機存取高層叢集、請先輸入「AZ登入」、然後輸入認證資料。
- 執行下列兩個命令：

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

- 在Azure CLI中輸入此命令：

```
kubectl get nodes
```



如果這六個節點都已啟動並執行、則您的高效能叢集已準備就緒並連線至本機環境。

```
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 ~ %
```

建立委派子網路**Azure NetApp Files** 以供使用

若要建立Azure NetApp Files 委派的子網路以供使用、請依照下列一系列步驟操作：

- 瀏覽至Azure入口網站內的虛擬網路。尋找您新建立的虛擬網路。它應該有一個前置詞、例如：在此處看到的；按一下虛擬網路的名稱。

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. 按一下子網路、然後從頂端工具列選取+子網路。

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

Search subnets

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

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

3. 在子網路委派標題下、輸入名稱「anf.sn」、然後選取Microsoft.NetApp/volumes。請勿變更任何其他項目。按一下「確定」。

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 ⓘ ▼

將實體磁碟區分配給應用程式叢集、並作為Kubernetes中的持續磁碟區宣告（PVCS）使用。Azure NetApp Files而這項配置則可讓我們靈活地將磁碟區對應到不同的服務、無論是Jupyter筆記型電腦、無伺服器功能等等

服務使用者可從平台使用多種儲存設備。此功能的主要優點Azure NetApp Files 包括：

- 提供使用者使用快照的能力。
- 可讓使用者將大量資料儲存在Azure NetApp Files 功能區上。
- 在Azure NetApp Files 大量檔案上執行機型時、請取得效能優勢。

系統設定Azure NetApp Files

若要完成Azure NetApp Files 設定、您必須先依照中所述進行設定 "[快速入門：設定Azure NetApp Files 功能以建立NFS磁碟區](#)"。

不過、您可以略過建立Azure NetApp Files NFS Volume以供使用的步驟、因為您將透過Trident建立Volume。在繼續之前、請確定您已：

- 1. ["已註冊Azure NetApp Files 為NetApp資源供應商（透過Azure Cloud Shell）"](#)。
- 2. ["在Azure NetApp Files 不執行任何作業的情況下建立帳戶"](#)。
- 3. ["設定容量資源池"](#)（視您的需求而定、最低4TiB標準或Premium）。

利用虛擬網路和**Azure NetApp Files** 虛擬化網路來對等化

接下來、Azure NetApp Files 請依照下列步驟、將使用者虛擬網路（vnet）與該虛擬網路（vnet）對等：

- 1. 在Azure入口網站頂端的搜尋方塊中、輸入虛擬網路。
- 2. 按一下vnet aks-vnet-name、然後在搜尋欄位中輸入「服務」。
- 3. 按一下「+新增」、然後輸入下表所提供的資訊：

欄位	價值或說明
對等連結名稱	aks-vnet-name_to_anf
訂閱ID	訂閱Azure NetApp Files 您所要登入的物件網
vnet對等合作夥伴	網版Azure NetApp Files



保留所有非星號區段的預設值

- 4. 按一下「新增」或「確定」、將對等新增至虛擬網路。

如需詳細資訊、請造訪 ["建立、變更或刪除虛擬網路對等關係"](#)。

Trident

Trident是NetApp為應用程式容器持續儲存所維護的開放原始碼專案。Trident已實作為外部資源配置程式控制器、以Pod本身的形式執行、監控磁碟區、並將資源配置程序完全自動化。

NetApp Trident可建立及附加持續容量、以儲存訓練資料集和訓練模型、順利與K8s整合。這項功能可讓資料科學家 and 資料工程師更輕鬆地使用K8s、而不需費心手動儲存和管理資料集。Trident也不需要資料科學家學習管理新的資料平台、因為它透過邏輯API整合來整合資料管理相關工作。

安裝Trident

若要安裝Trident軟體、請完成下列步驟：

- 1. ["第一次安裝Helm"](#)。
- 2. 下載並解壓縮Trident 21.01.1安裝程式。

```
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. 將目錄變更為「Trident安裝程式」。

```
cd trident-installer
```

4. 將「tridentctl」複製到系統「\$path」中的目錄

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

5. 使用Helm在K8s叢集上安裝Trident：

- a. 將目錄變更為helm目錄。

```
cd helm
```

- b. 安裝Trident。

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

- c. 以一般的K8s方法檢查Trident Pod的狀態：

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

- d. 如果所有的Pod都已啟動且正在執行、則會安裝Trident、您可以繼續向前邁進。

設定**Azure NetApp Files** 不中斷的後端與儲存類別

若要設定Azure NetApp Files 不完整的後端與儲存類別、請完成下列步驟：

1. 切換回主目錄。

```
cd ~
```

2. 複製 **"專案儲存庫"** 「lane detection-SCNN-Horovod」。
3. 移至「trident組態」目錄。

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

4. 建立Azure服務原則（服務原則是Trident如何與Azure通訊以存取Azure NetApp Files 您的整套資源）。

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

輸出應如下所示：

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

5. 建立Trident的「後端json」檔案。
6. 使用偏好的文字編輯器、從「anf-backend.json」檔案中的下表填寫下列欄位。

欄位	價值
訂閱ID	您的Azure訂閱ID
TenantId	您的Azure租戶ID（上一步AZ廣告服務輸出）
ClientId	您的應用程式ID（從上一步AZ廣告服務輸出）
用戶端機密	您的密碼（取自上一步AZ廣告服務的輸出）

檔案應如下所示：


```
{
  "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"
  }
}
```

7. 指示Trident在Azure NetApp Files 「Trident」命名空間中建立「支援」後端、使用「anf-backend.json」做為組態檔、如下所示：

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

8. 建立儲存類別：

- a. K8使用者使用以名稱指定儲存類別的PVCS來配置磁碟區。指示K8s建立儲存類別「azurenetafiles」、以參照Azure NetApp Files 上一步建立的「背後」、使用下列項目：

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

- b. 使用下列命令檢查是否已建立儲存類別：

```
kubectl get sc azurenetafiles
```

輸出應如下所示：

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

在高性能上部署及設定**Volume Snapshot**元件

如果叢集未預先安裝正確的Volume Snapshot元件、您可以執行下列步驟、手動安裝這些元件：



若為AKS 1.18.14、則不會預先安裝Snapshot控制器。

1. 使用下列命令安裝Snapshot Beta客戶需求日：

```
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提供的下列文件來安裝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. 設定K8s「volumesnapshotClass」：在建立Volume Snapshot之前、請先 "[Volume Snapshot類別](#)" 必須設定。建立適用於Azure NetApp Files 功能不全的Volume Snapshot類別、並使用NetApp Snapshot技術來達到ML版本管理。建立「volumesnapshotClass NetApp-csi快照類別」、並將其設為預設的「volumesnapshotClass」、例如：

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

輸出應如下所示：

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

4. 使用下列命令檢查是否已建立Volume Snapshot複本類別：

```
kubectl get volumesnapshotclass
```

輸出應如下所示：

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

執行：AI安裝

若要安裝RUN：AI、請完成下列步驟：

1. "安裝RUN：AI叢集於AKS上"。
2. 前往app.runai.ai、按一下「Create New Project（建立新專案）」、然後將其命名為「lane detection（線道偵測）」它將在K8s叢集上建立命名空間、開頭為「Runai」、後面接著專案名稱。在這種情況下、建立的命名空間將會是Runai-lane偵測。

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. "安裝RUN：AI CLI"。
4. 在終端機上、使用下列命令將lane偵測設為預設執行：AI project：

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

輸出應如下所示：

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

5. 為專案命名空間（例如「lane detection」）建立ClusterRole和Cluster勞力 綁定、因此屬於「Runae-lane detection」命名空間的預設服務帳戶、在工作執行期間有權執行「volumesnapshot」作業：
 - a. 使用以下命令列出命名空間、檢查是否存在「Runae-lane偵測」：

```
kubectl get namespaces
```

輸出應如下所示：

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

6. 使用下列命令建立ClusterRole「netappsnapshot (netappsnapshot)」、和Cluster勞力 綁定「netappsnapshot (netappsnapshot)」：

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

下載並處理**TuSimple**資料集、做為**RUN：AI**工作

下載和處理TuSimple資料集的流程為RUN：AI工作是選用的。其中包括下列步驟：

1. 建置並推送Docker映像檔、或是如果您想要使用現有的Docker映像檔（例如「muneer7589/download-tuSimple:1.0」）、請省略此步驟
 - a. 切換至主目錄：

```
cd ~
```

- b. 前往「lane detection-SCNN-Horovod」專案的資料目錄：

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

- c. 修改「build」（建置）「image」（映像）「sh」（sh）Shell指令碼、並將Docker儲存庫變更為您的。例如、將「muneer7589」取代為您的泊塢視窗儲存庫名稱。您也可以變更泊塢視窗的影像名稱和標記（例如「下載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. 執行指令碼以建立泊塢視窗映像、並使用下列命令將其推送到泊塢視窗儲存庫：

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

2. 提交RUN：AI工作以下載、擷取、預先處理及儲存TuSimple lane偵測資料集至「PVC」、這是由NetApp Trident動態建立的：

- a. 使用下列命令提交RUN：AI工作：

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

- b. 輸入下表中的資訊、以提交RUN：AI工作：

欄位	價值或說明
-name	工作名稱
-PVC	格式為[StorageClassName]的PVc：大小：ContainerMountPath在上述工作提交中、您將使用Trident搭配儲存類別azurenetaappFiles、根據需求建立一個PVc。持續磁碟區容量為100Gi、安裝於路徑/mnt.
映像	建立此工作的容器時要使用的Docker影像

輸出應如下所示：

```
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. 列出提交的RUN：AI工作。

```
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 veronmartina 0 (0)
```

d. 檢查提交的工作記錄。

```
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. 列出所建立的「PVC」。請使用這個「PVC」命令進行下一步的訓練。

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

輸出應如下所示：

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

a. 檢查執行中的工作：AI UI（或「app.run.ai」）。

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

使用Horovod執行分散式線道偵測訓練

使用Horovod進行分散式通道偵測訓練是一項選擇性程序。不過、以下是相關步驟：

1. 建置並推送泊塢視窗映像、或是如果您想要使用現有的泊塢視窗映像（例如「muneer7589/der-lane detection:3.1」）、請跳過此步驟
 - a. 切換到主目錄。

```
cd ~
```

- b. 轉到專案目錄「lane detection-SCNN-Horovod.」

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

- c. 修改「build」（建置）「image」（映像）「sh」（sh）Shell指令碼、並將泊塢視窗儲存庫變更為您的（例如、將「muneer7589」取代為您的泊塢視窗儲存庫名稱）。您也可以變更泊塢視窗的影像名稱和標記（例如「dist-lane 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. 執行指令碼以建立泊塢視窗映像、然後推送至泊塢視窗儲存庫。

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

2. 提交RUN：AI工作以執行分散式訓練（MPI）：

- 使用提交執行：AI在上一步（用於下載資料）自動建立永久虛擬基礎架構、僅允許您存取Rwo、這不允許多個Pod或節點存取相同的永久虛擬基礎架構以進行分散式訓練。將存取模式更新為ReadWriteMany、並使用Kubernetes修補程式來執行此作業。
- 首先、請執行下列命令來取得PVC的Volume名稱：

```
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
```

- 修補磁碟區、並將存取模式更新為ReadWriteMany（以下列命令取代Volume名稱）：

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

- 提交RUN：AI MPI工作、以便使用下表中的資訊來執行分散式訓練工作：

```

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"

```

欄位	價值或說明
名稱	分散式訓練工作的名稱
大型shm	掛載大型的開發/ shm裝置這是安裝在RAM上的共享檔案系統、提供足夠大的共享記憶體、讓多個CPU工作者能夠處理批次並將其載入CPU RAM。
程序	分散式訓練程序的數量
GPU	要分配給此工作的GPU /程序數目、有三個GPU工作程序 (--Processes=3)、每個都分配一個GPU (-GPU 1)
PVC	使用先前工作 (download-tuSimple資料) 所建立的現有持續磁碟區 (PVC-download-tuSimple、data、PVC-download-tue-tuSimple)、並安裝在路徑/mnt
映像	建立此工作的容器時要使用的Docker影像
定義要在容器中設定的環境變數	
使用工作者	將引數設為true會開啟多重程序資料載入
員工人數	資料載入器工作程序的數目
批次大小	訓練批次大小
US_VAL	將引數設為true可進行驗證
Val_batch_size	驗證批次大小
啟用快照	將引數設為true可取得資料和訓練模型快照、以利ML版本管理
PVC_name	要擷取快照的Pvc名稱。在上述提交工作時、您將取得PVC-download-tuSimple資料0的快照、其中包含資料集和訓練模型

輸出應如下所示：

```
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. 列出已提交的工作。

```
runai list jobs
```

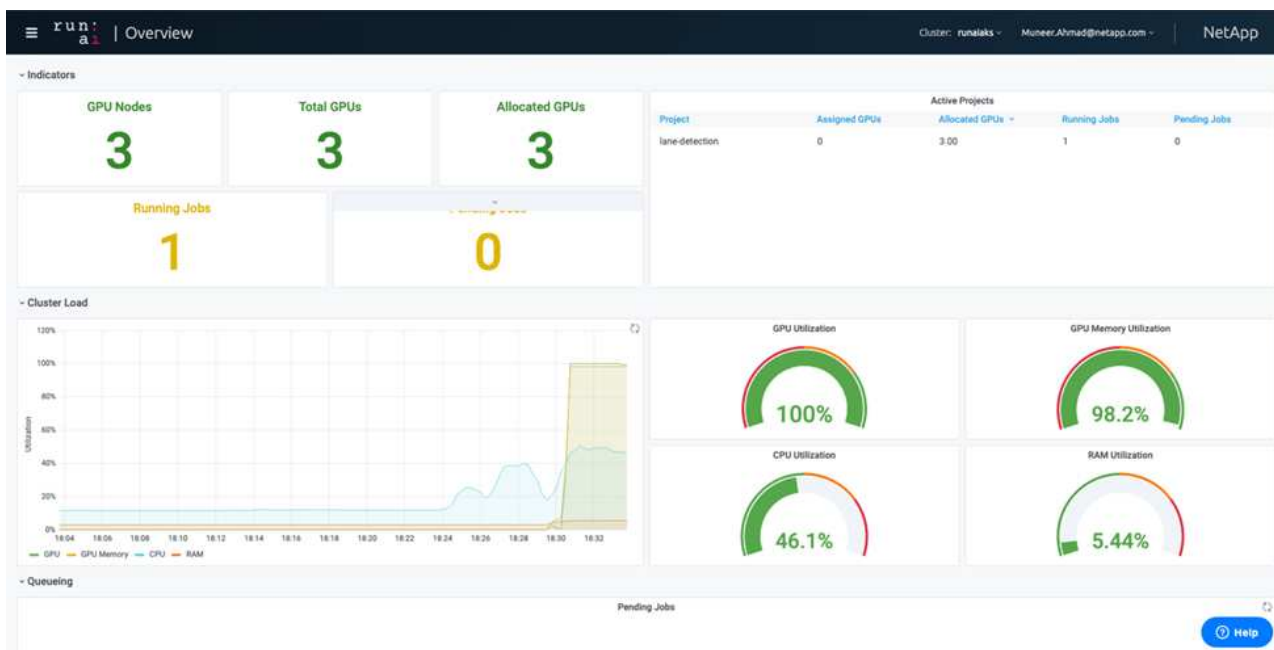
NAME	SERVICE URL(S)	STATUS	AGE	NODE	IMAGE	TYPE	PROJECT	USER	GPUs Allocated (Requested)	PODs
download-tusimple-data		Succeeded	1d		muneer7589/download-tusimple:1.0	Train	lane-detection	verronmartina	0 (0)	0 (0)
dist-lane-detection-training		Init:0/1	2m	<multiple>	muneer7589/dist-lane-detection:3.1	Train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

f. 提交的工作記錄：

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

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

g. 請在RUN（執行）中檢查訓練工作：AI GUI（或app.runai.ai): RUN：AI儀表板）、如下圖所示。第一張圖詳細說明分配給分散式訓練工作的三個GPU、分別位於下列三個節點上、以及第二個RUN：AI工作：



The screenshot shows the RunAI Jobs dashboard. On the left, a table lists jobs with columns: Job Name, Status, User, Project, and Total Run Time. The first job, 'dist-lane-detection-training', is highlighted with a 'Running' status. On the right, a detailed view of this job is shown, including a 'Status' bar (Running) and a 'GPUs' tab. The GPUs tab displays a table with columns: GPU, Node, Utilization, GPU Memory, and Used GPU Memory. It shows three GPUs, each with 100% utilization and 15.9 GB of memory used out of 15.61 GB.

h. 訓練完成後、請查看已建立並連結RUN : AI job.的NetApp Snapshot複本。

```
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
```

從NetApp Snapshot複本還原資料

若要從NetApp Snapshot複本還原資料、請完成下列步驟：

1. 切換到主目錄。

```
cd ~
```

2. 轉到項目目錄"lane detection-SCNN-Horovod"。

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

3. 修改「REstore-snapshot-PVC.yaml」、並將「data來源」「名稱」欄位更新為您要從中還原資料的Snapshot複本。您也可以變更要將資料還原到的PVC名稱、例如「restore-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-PVC.yaml」建立新的PVC。

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

輸出應如下所示：

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

5. 如果您想要使用剛還原的資料進行訓練、則提交工作內容與之前相同；提交訓練工作時、只能以還原的「PVC_name」取代「PVC_name」、如下列命令所示：

```

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"

```

效能評估

為了顯示解決方案的線性擴充性、我們針對兩種情境進行了效能測試：一種GPU和三種GPU。GPU配置、GPU和記憶體使用率、在TuSimple lane偵測資料集的訓練中、已擷取不同的單節點和三節點測量數據。資料增加五倍、只是為了在訓練過程中分析資源使用率。

此解決方案可讓客戶從小型資料集和幾個GPU開始著手。當資料量和GPU需求增加時、客戶可以在標準層中動態橫向擴充TB、並快速擴充至頂級層、以獲得每TB 4倍的處理量、而無需移動任何資料。本節將進一步說明此

程序： "服務層級Azure NetApp Files"。

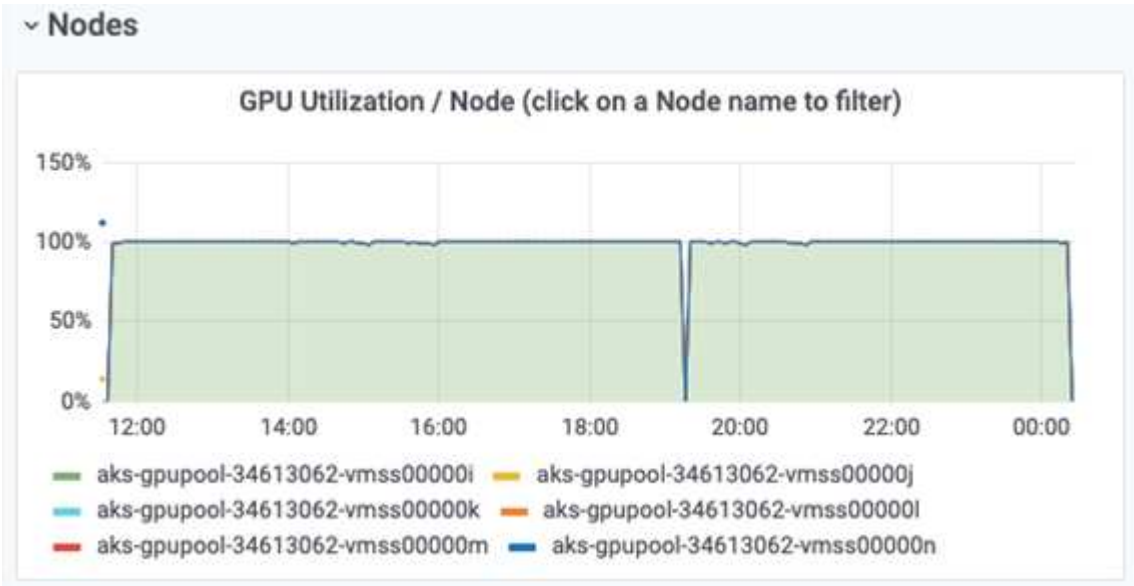
單一GPU的處理時間為12小時45分鐘。三個節點上的三個GPU處理時間約為4小時30分鐘。

本文件其餘部分所顯示的數字、說明根據個別業務需求而提供的效能與擴充性範例。

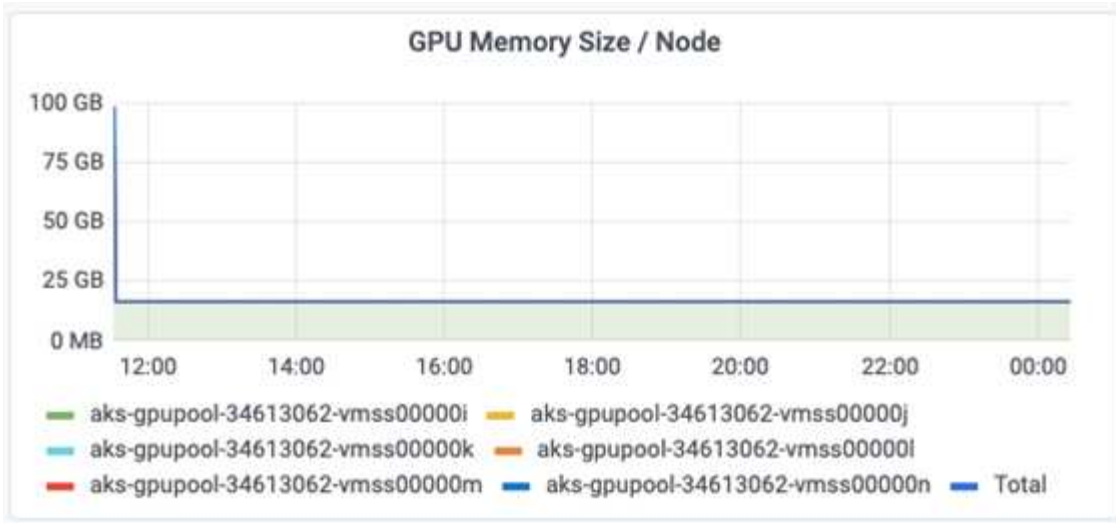
下圖說明1 GPU配置與記憶體使用率。



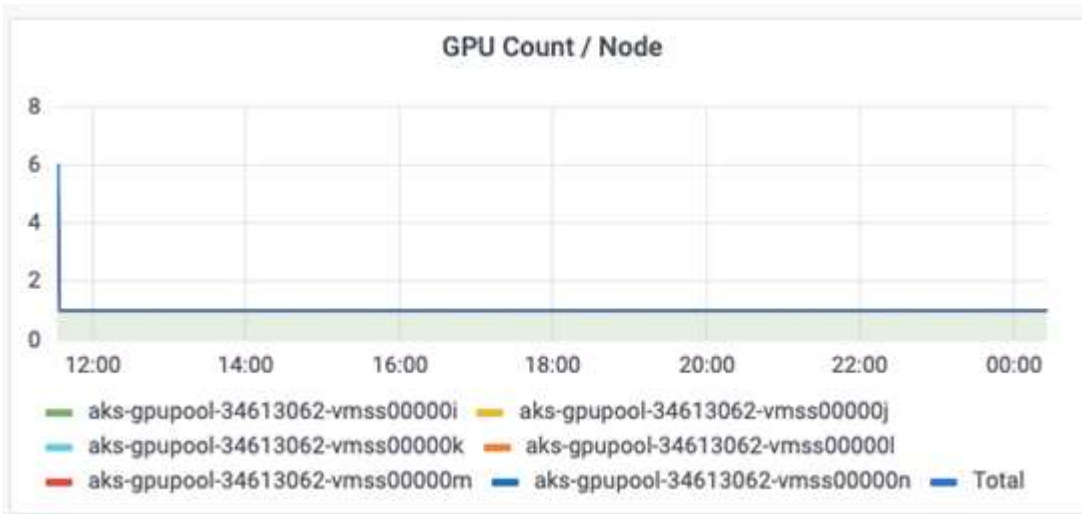
下圖說明單一節點GPU使用率。



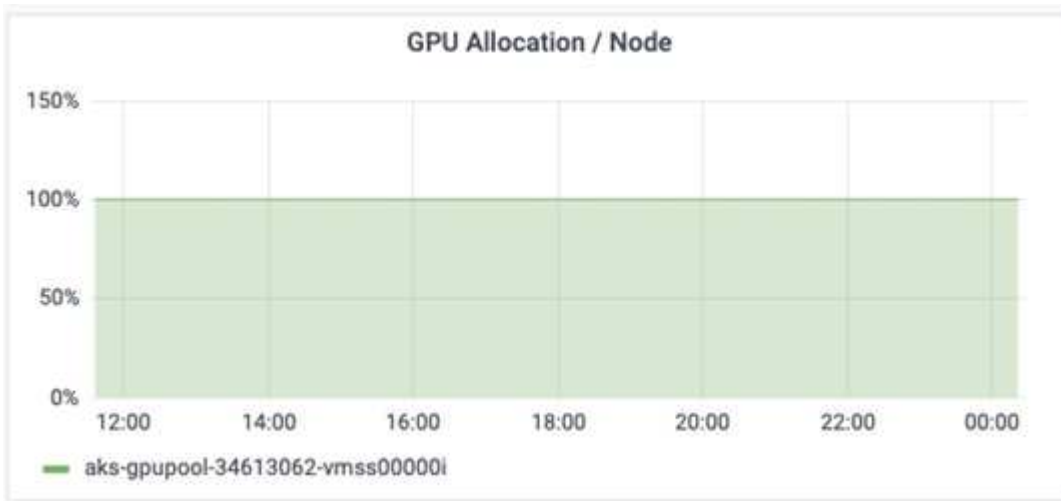
下圖說明單一節點記憶體大小（16GB）。



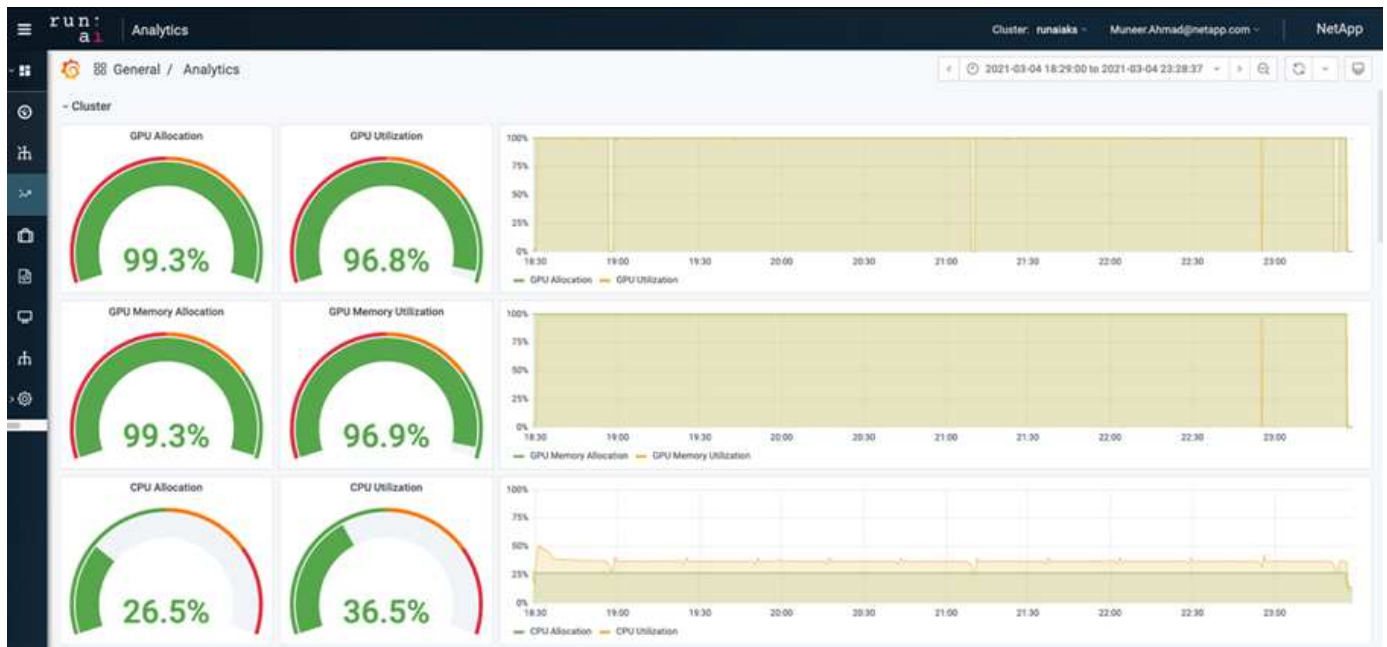
下圖說明單一節點GPU數（1）。



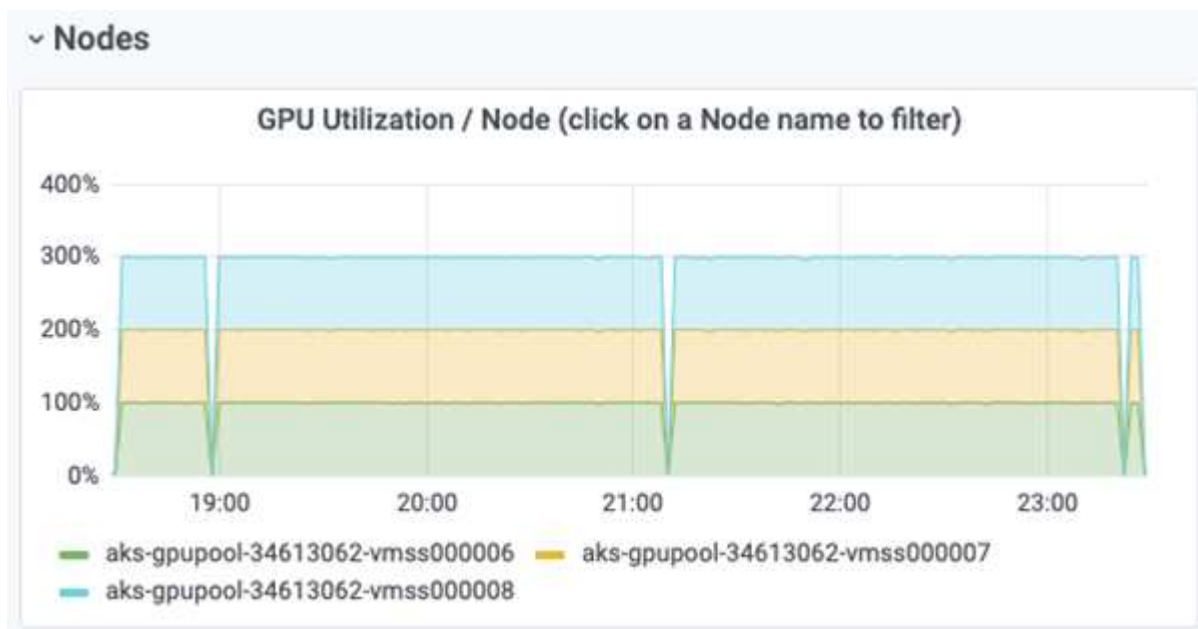
下圖說明單一節點GPU配置（%）。



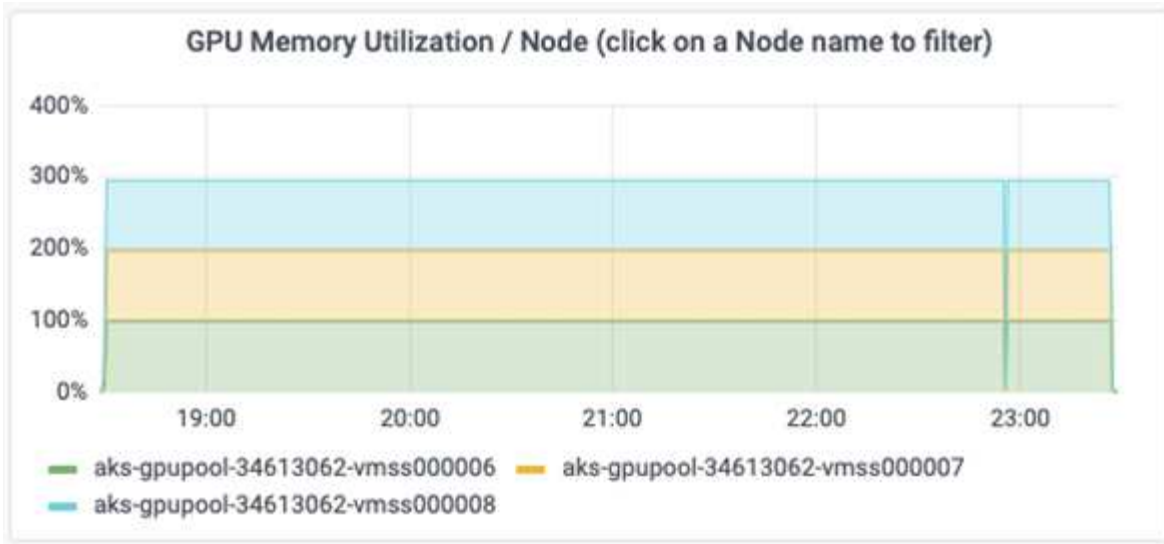
下圖說明三個節點的三個GPU：GPU配置與記憶體。



下圖說明三個節點使用率的三個GPU（%）。◦



下圖說明三個節點的三個GPU記憶體使用率（%）。◦



服務層級 Azure NetApp Files

您可以將磁碟區移至另一個使用的容量集區、以變更現有磁碟區的服務層級 **"服務層級"** 您想要的磁碟區。此磁碟區現有的服務層級變更不需要移轉資料。這也不會影響對磁碟區的存取。

動態變更磁碟區的服務層級

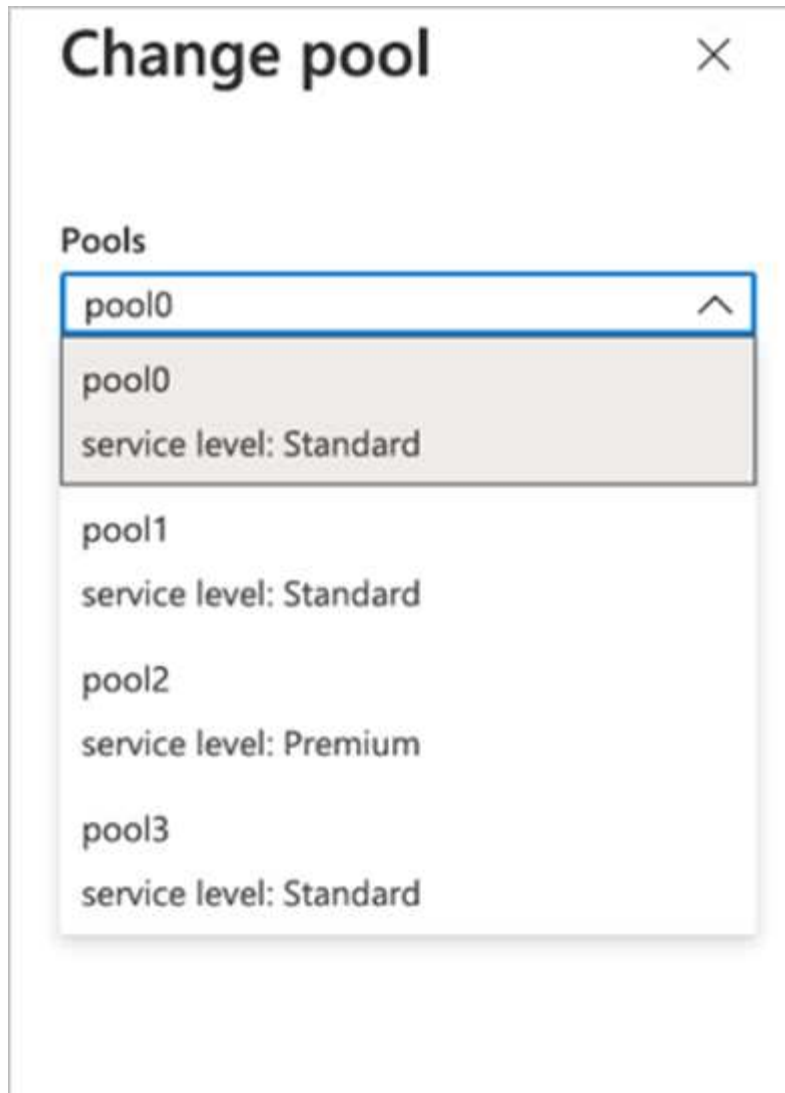
若要變更 Volume 的服務層級、請執行下列步驟：

1. 在「Volumes（磁碟區）」頁面上、以滑鼠右鍵按一下您要變更其服務層級的磁碟區。選取變更資源池。

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「變更資源池」視窗中、選取您要將磁碟區移至的容量資源池。然後按一下「OK（確定）」。



自動化服務層級變更

動態服務層級變更目前仍在「公開預覽」中、但預設不會啟用。若要在Azure訂閱上啟用此功能、請依照文件中提供的步驟執行 ["動態變更磁碟區的服務層級"](#)。」

- 您也可以針對Azure使用下列命令：CLI。如需變更Azure NetApp Files 資源池大小的詳細資訊、請造訪 ["AZ netappFiles Volume：管理Azure NetApp Files fz（anf）Volume資源"](#)。

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

- 此處顯示的「set-aznetappfilesvolume pool」指令程式可變更Azure NetApp Files 一個現象區的集區。如需變更Volume Pool大小和Azure PowerShell的詳細資訊、請參閱 ["變更Azure NetApp Files 適用於某個需求量的資源池"](#)。

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

結論

NetApp與RUN：AI合作建立本技術報告、以展示Azure NetApp Files 與眾不同的功能、搭配RUN：AI平台、簡化AI工作負載的協調作業。本技術報告提供參考架構、可簡化資料管線和工作負載協調的流程、以利分散式通道偵測訓練。

最後、關於大規模分散式訓練（尤其是公有雲環境）、資源協調與儲存元件是解決方案的關鍵部分。確保資料管理不會阻礙多個GPU處理、因此能達到GPU週期的最佳使用率。如此一來、系統就能以最具成本效益的方式進行大規模的分散式訓練。

NetApp所提供的Data Fabric可讓資料科學家和資料工程師在內部部署和雲端之間建立連線、以取得同步資料、而無需執行任何手動介入、克服了這項挑戰。換句話說、資料架構可順暢地管理分散在多個位置的AI工作流程。此外、它也能將資料帶離運算近、並在需要時隨時隨地執行分析、訓練和驗證、進而提升隨需資料的可用度。這項功能不僅能實現資料整合、也能保護和保護整個資料傳輸途徑。

其他資訊

若要深入瞭解本文所述資訊、請檢閱下列文件和 / 或網站：

- 資料集：TuSimple

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

- 深度學習網路架構：空間凸狀神經網路

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

- 分散式深度學習訓練架構：Horovod

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

- RUN：AI Container協調解決方案：RUN：AI產品簡介

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

- 執行：AI安裝文件

["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RUN（執行）中提交工作：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 Cloud資源Azure NetApp Files

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

- Azure Kubernetes服務

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

- Azure VM SKU

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

- 搭載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)

- 採用NetApp技術的Data Fabric

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

- NetApp 產品文件

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

TR-4841：混合雲AI作業系統、含資料快取

Rick Huang、David Arnette、NetApp Yochay Ettun、cnvrg-io

資料爆炸性成長、以及ML和AI的指數成長、已融合在一起、創造出一個具有獨特開發與實作挑戰的字節經濟。

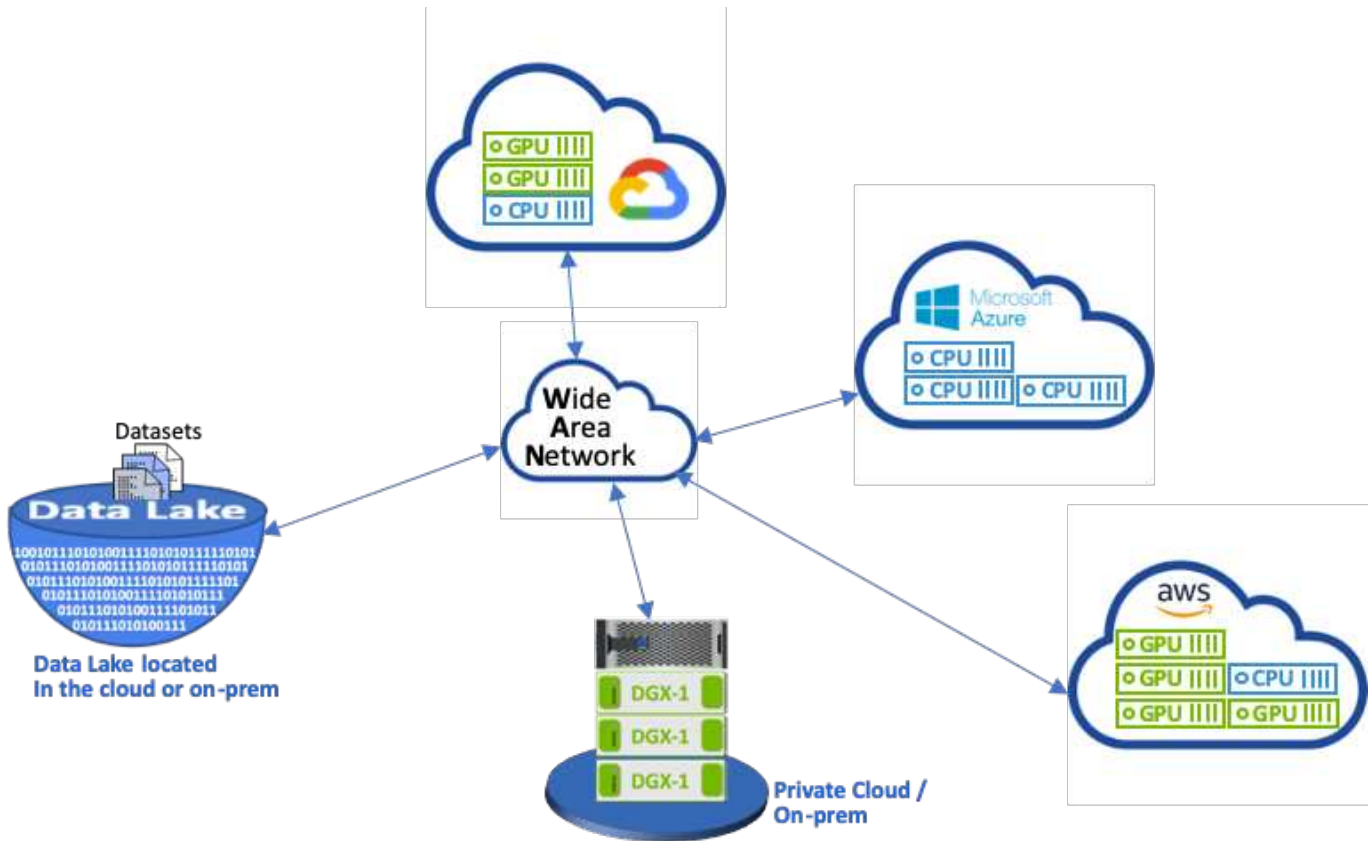
雖然大家都知道、ML模型需要大量資料、而且需要近端的高效能資料儲存設備來處理運算資源、但實際上實作這種模式並不太直接、尤其是混合雲和彈性運算執行個體。大量資料通常儲存在低成本的資料湖中、因為GPU等高效能AI運算資源無法有效存取資料。在混合雲基礎架構中、有些工作負載會在雲端上運作、有些工作負載則位於內部部署環境或完全位於不同的HPC環境中、這種情況更形嚴重。

在本文件中、我們提供一款新穎的解決方案、讓IT專業人員和資料工程師能夠建立真正的混合雲AI平台、並具備拓撲感知資料中心、讓資料科學家能夠在運算資源附近、立即自動建立資料集快取、無論位於何處。因此、不僅能完成高效能模式訓練、還能創造更多效益、包括多位AI從業人員的協同作業、他們可以立即存取資料集版本中樞內的資料集快取、版本和線路。

使用案例總覽與問題陳述

資料集與資料集版本通常位於資料湖中、例如NetApp StorageGRID 以物件為基礎的儲存

設備、可降低成本及提供其他營運優勢。資料科學家會將這些資料集拉出、並以多個步驟來進行設計、以準備好使用特定模型進行訓練、通常會在過程中建立多個版本。下一步、資料科學家必須挑選最佳化的運算資源（GPU、高階CPU執行個體、內部部署叢集等）來執行模型。下圖說明ML運算環境中資料集的鄰近度不足。



然而、多項訓練實驗必須在不同的運算環境中平行執行、每項都需要從資料湖下載資料集、這是一項昂貴且耗時的程序。無法保證資料集與運算環境的距離（尤其是混合雲）。此外、在同一個資料集上執行自己實驗的其他團隊成員、也必須經歷同樣艱鉅的程序。除了明顯緩慢的資料存取速度之外、還有難以追蹤資料集版本、資料集共用、協同作業和可重複性等挑戰。

客戶需求

客戶的需求可能會有所不同、以便在有效率地使用資源的情況下執行高效能ML；例如、客戶可能需要下列項目：

- 從執行訓練模式的每個運算執行個體快速存取資料集、而不會產生昂貴的下載和資料存取複雜度
- 在雲端或內部部署中使用任何運算執行個體（GPU或CPU）、而不需擔心資料集的位置
- 在同一個資料集上同時執行多項訓練實驗、並使用不同的運算資源、而不會產生不必要的延遲和資料延遲、進而提升效率和生產力
- 將運算執行個體成本降至最低
- 利用工具來記錄資料集、其資料類型、版本及其他中繼資料詳細資料、藉此改善可重複性
- 增強共享與協同作業、讓團隊中的任何授權成員都能存取資料集並執行實驗

若要使用NetApp ONTAP 支援資料集快取管理軟體來實作資料集快取、客戶必須執行下列工作：

- 設定和設定最接近運算資源的NFS儲存設備。
- 判斷要快取的資料集和版本。
- 監控已認可給快取資料集的總記憶體、以及可用於其他快取認可的NFS儲存容量（例如快取管理）。
- 如果資料集在某段時間內未使用、則會在快取中逾時。預設值為一天、其他組態選項則可供使用。

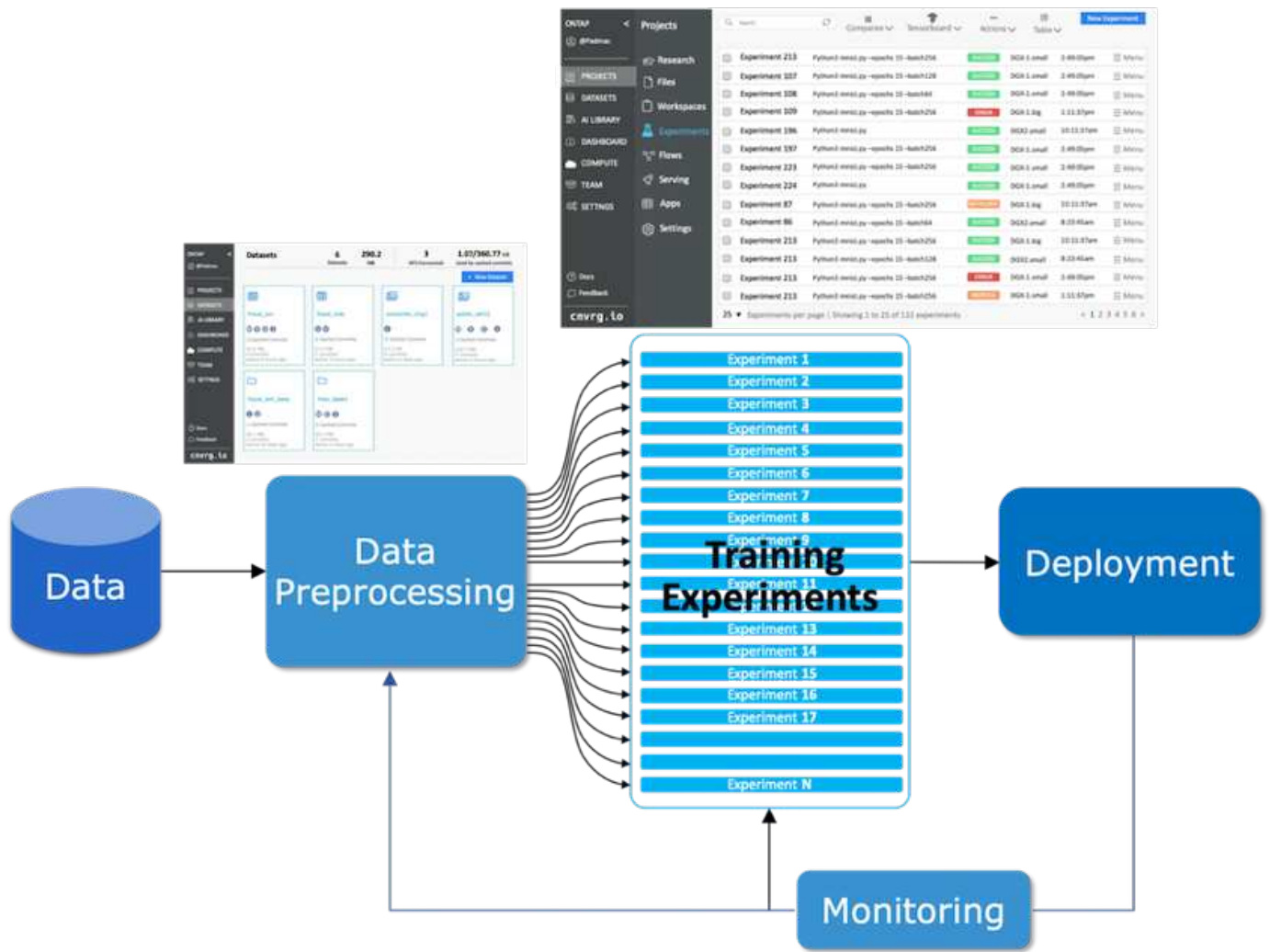
解決方案總覽

本節將回顧傳統的資料科學管道及其缺點。同時也介紹建議的資料集快取解決方案架構。

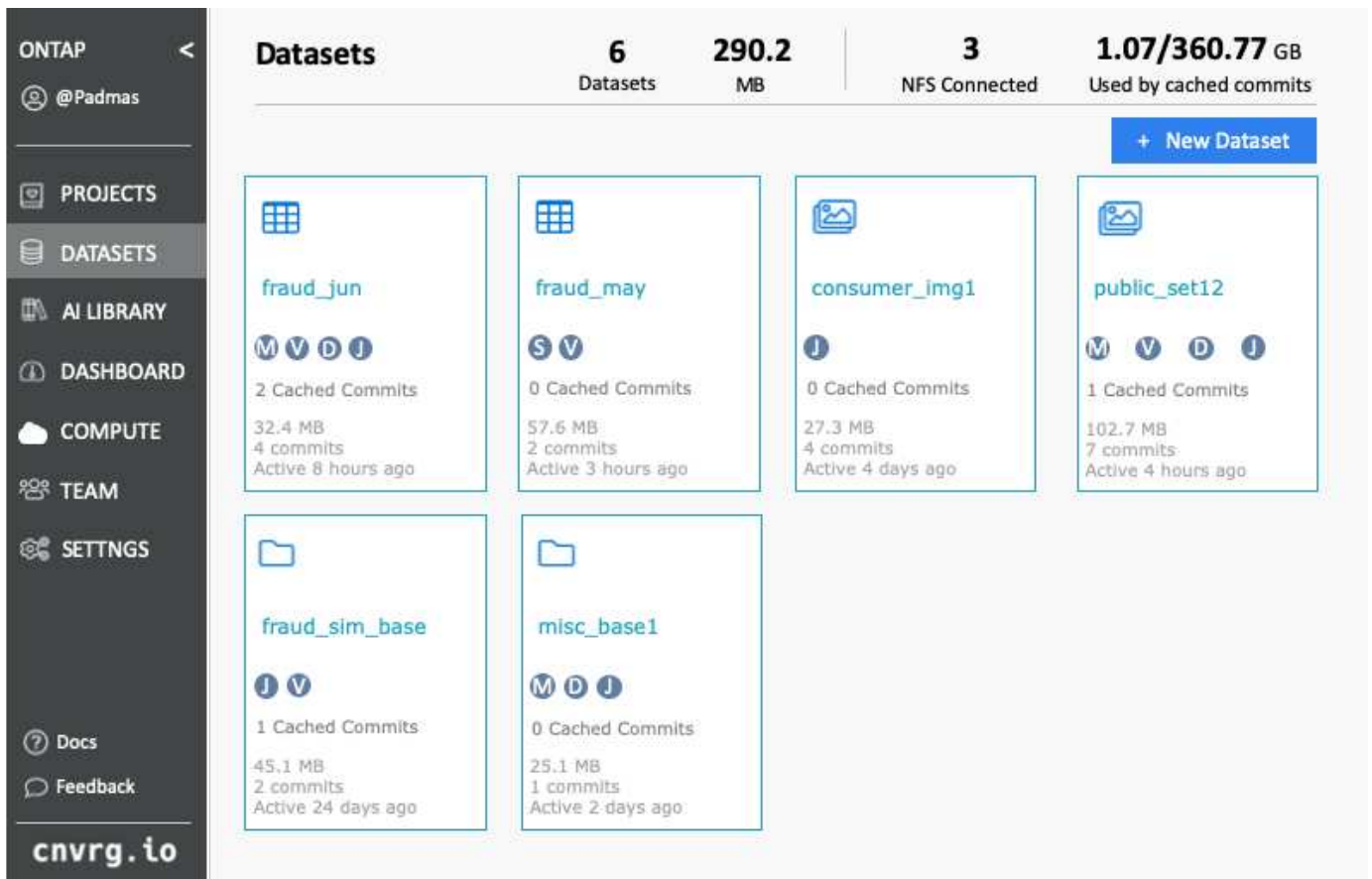
傳統的資料科學管道與缺點

典型的ML模型開發與部署順序涉及迭代步驟、包括下列步驟：

- 擷取資料
- 資料預先處理（建立多個版本的資料集）
- 執行多項涉及超參數最佳化、不同模型等的實驗
- 部署
- 監控cnvrg-IO已開發出全方位平台、可將研究到部署等所有工作自動化。下圖顯示與管線相關的儀表板快照範例。



在公有儲存庫和私有資料中、經常會有多個資料集在活動中。此外、每個資料集可能會因為資料集清理或功能工程而產生多個版本。需要提供資料集線器和版本集線器的儀表板、以確保團隊能夠使用協同作業和一致性工具、如下圖所示。



下一步是訓練、訓練模式需要多個平行執行個體、每個執行個體都與資料集和特定運算執行個體相關聯。將資料集繫結至特定運算執行個體的特定實驗、是一項挑戰、因為某些實驗可能是由Amazon Web Services (AWS) 的GPU執行、而其他實驗則由內部部署的DGX-1或DGX-2執行個體執行。在GCP的CPU伺服器上執行其他實驗、而資料集位置與執行訓練的運算資源並不合理。合理的鄰近範圍、可將資料集儲存設備與運算執行個體之間的完整10GbE或更低延遲連線。

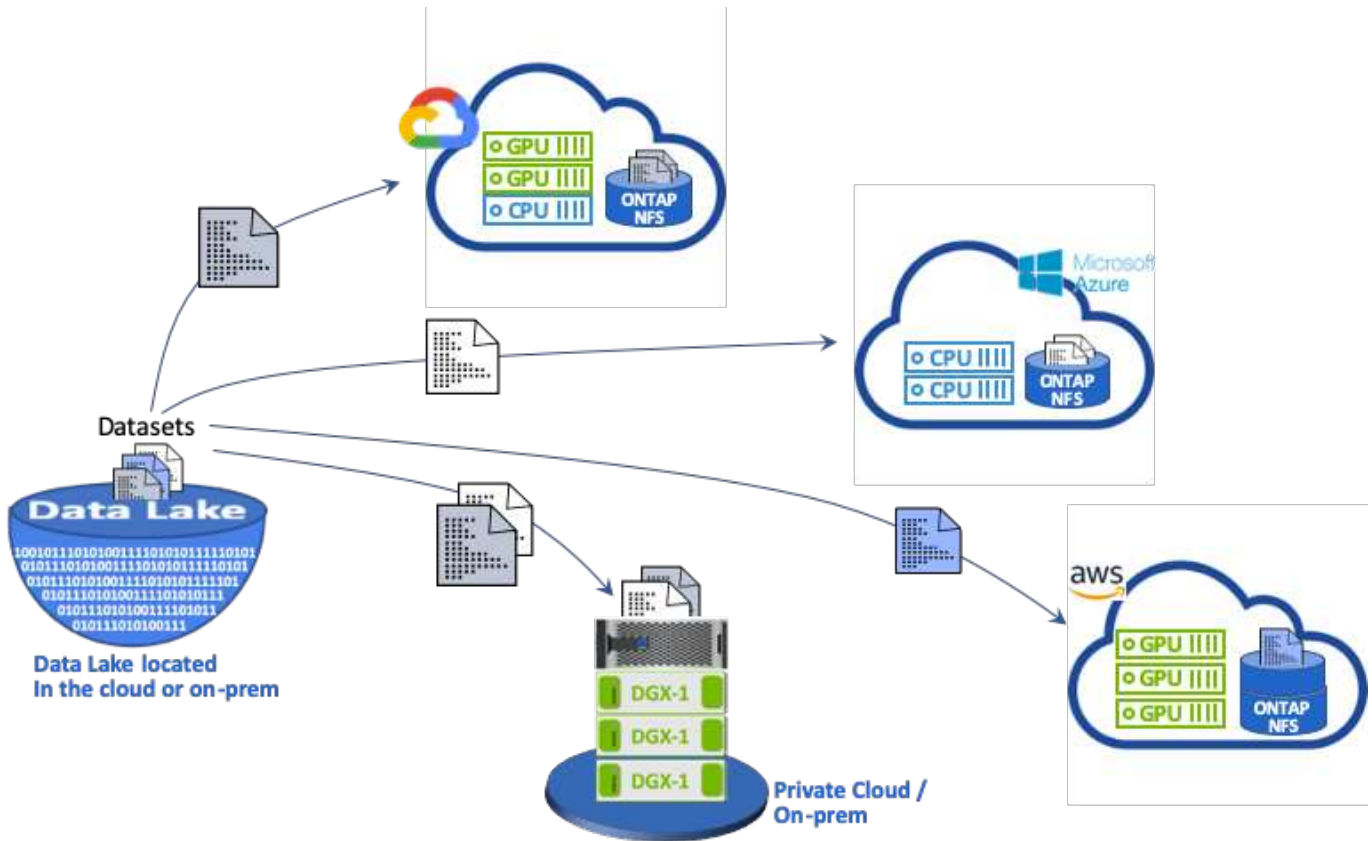
資料科學家通常會將資料集下載至執行訓練和執行實驗的運算執行個體。不過、這種方法可能有幾個問題：

- 當資料科學家將資料集下載至運算執行個體時、並不保證整合式運算儲存設備具備高效能（高效能系統的範例是ONTAP AFF 以支援A800 NVMe解決方案為例）。
- 當下載的資料集位於單一運算節點時、分散式模型在多個節點上執行時、儲存設備可能會成為瓶頸（與NetApp ONTAP 的高效能分散式儲存設備不同）。
- 由於佇列衝突或優先順序、訓練實驗的下一次迭代作業可能會在不同的運算執行個體中執行、同樣也會從資料集到運算位置建立相當長的網路距離。
- 在同一個運算叢集上執行訓練實驗的其他團隊成員無法共用此資料集；每個團隊成員都會從任意位置執行（昂貴）資料集下載。
- 如果後續訓練工作需要相同資料集的其他資料集或版本、資料科學家必須重新執行（昂貴）資料集下載、將資料集下載至執行training.NetApp的運算執行個體、並執行cnvrg-.IO建立新的資料集快取解決方案、以消除這些障礙。此解決方案可將Hot資料集快取至ONTAP 高效能的儲存系統、以加速執行ML管線。使用支援NetApp的Data Fabric（例如、Re A800）、資料集會在運算組合的資料架構中快取一次（而且只快取一次）ONTAP AFF。由於NetApp ONTAP 不間斷NFS高速儲存設備可支援多個ML運算節點、因此訓練模式的效能已經過最佳化、可為組織帶來成本節約、生產力和營運效率。

解決方案架構

此解決方案來自NetApp和cnvrg-IO、提供資料集快取功能、如下圖所示。資料集快取可讓資料科學家挑選所需的資料集或資料集版本、並將其移至ONTAP 靠近ML運算叢集的支援NFS快取。資料科學家現在可以執行多項實驗、而不會產生延遲或下載。此外、所有協同作業的工程師都能將相同的資料集用於附加的運算叢集（可自由選擇任何節點）、而無需從資料湖下載額外資料。提供資料科學家儀表板、可追蹤及監控所有資料集和版本、並提供快取資料集的檢視。

cnvrg-IO平台會自動偵測未在特定時間內使用的老舊資料集、並從快取中予以移出、因為快取會為較常用的資料集保留可用的NFS快取空間。請務必注意ONTAP、使用效益技術的資料集快取功能可在雲端和內部部署中運作、因此能提供最大的靈活度。



概念與元件

本節涵蓋與ML工作流程中的資料快取相關的概念與元件。

機器學習

對於全球許多企業和組織而言、ML正迅速成為不可或缺的一環。因此、IT與DevOps團隊現在面臨著將ML工作負載標準化、以及配置雲端、內部部署與混合式運算資源的挑戰、這些資源可支援ML工作與管線所需的動態密集工作流程。

以Container為基礎的機器學習與Kubernetes

容器是獨立的使用者空間執行個體、可在共享主機作業系統核心上執行。容器的採用率迅速增加。Container提供許多與虛擬機器（VM）相同的應用程式沙箱效益。不過、由於虛擬機器所仰賴的Hypervisor和客體作業系統層已經被淘汰、因此容器的重量遠較輕。

容器也能直接透過應用程式、有效地封裝應用程式相依性、執行時間等項目。最常用的容器包裝格式是Docker容器。以Docker Container格式容器化的應用程式、可在任何能夠執行Docker Container的機器上執行。即使應用程式的相依性並不存在於機器上、這也是如此、因為所有相依性都會封裝在容器本身中。如需詳細資訊、請參閱 ["Docker網站"](#)。

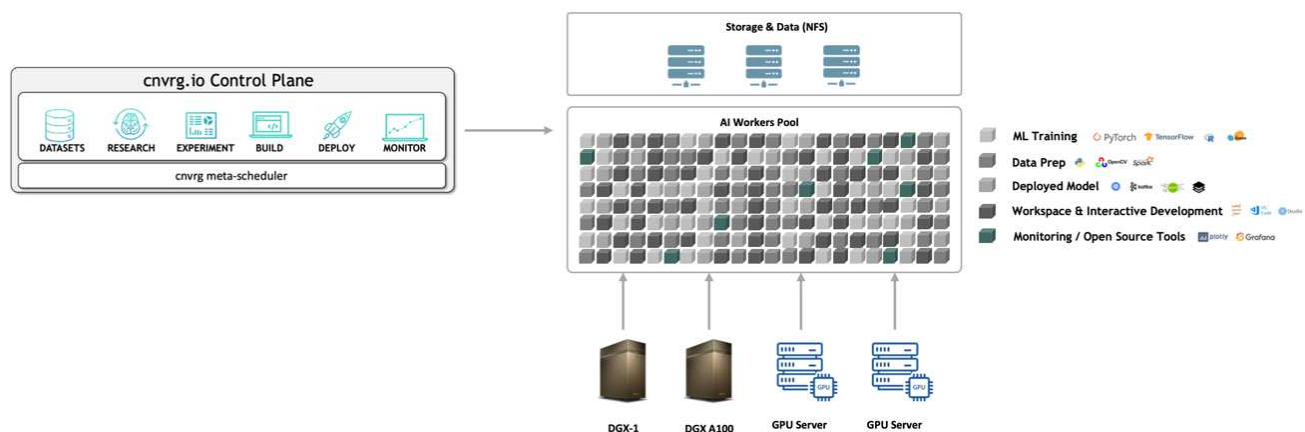
廣受歡迎的Container Orchestrator Kubernetes可讓資料科學家啟動靈活、以容器為基礎的工作和管線。此外、基礎架構團隊也能在單一託管與雲端原生環境中、管理及監控ML工作負載。如需詳細資訊、請參閱 ["Kubernetes網站"](#)。

cnvrg-io

Cnvrg-IO是一套AI作業系統、可將企業管理、擴充及加速AI與資料科學的開發、從研究到正式作業的方式轉變成全新的方式。程式碼優先平台是由資料科學家為資料科學家所建置、可靈活地在內部部署或雲端上執行。利用模型管理、MLOps和持續的ML解決方案、cnvrg-IO將頂尖技術引進資料科學團隊、讓他們能將更少的時間花在DevOps上、專注於真正的魔力演算法。自從使用cnvrg-IO之後、各產業的團隊已獲得更多的生產模式、進而提高商業價值。

Cnvrg-IO中繼排程器

Cnvrg：IO具有獨特的架構、可讓IT和工程師將不同的運算資源附加至同一個控制面板、並讓cnvrg-IO管理所有資源中的ML工作。這表示它可以附加多個內部部署的Kubernetes叢集、VM伺服器及雲端帳戶、並在所有資源上執行ML工作負載、如下圖所示。



Cnvrg-IO資料快取

Cnvrg-IO可讓資料科學家利用其資料快取技術來定義冷熱資料集版本。根據預設、資料集會儲存在集中式物件儲存資料庫中。然後、資料科學家可以快取所選運算資源上的特定資料版本、以節省下載時間、進而提高ML開發與生產力。快取且未使用數天的資料集會自動從選取的NFS清除。快取和清除快取只要按一下滑鼠、不需要編碼、IT或DevOps工作。

Cnvrg-IO流程和ML管路

Cnvrg-IO流程是建置正式作業ML管線的工具。流程中的每個元件都是在使用基礎泊塢視窗映像的選定運算上執行的指令碼/程式碼。這項設計可讓資料科學家和工程師建立單一管線、同時在內部部署和雲端上執行。Cnvrg-IO可確保資料、參數和成品在不同的元件之間移動。此外、系統會監控並追蹤每個流程、以確保100%可重現的資料科學。

Cnvrgr-I/O核心

Cnvrgr-I/O核心是資料科學社群的免費平台、可協助資料科學家更專注於資料科學、而非DevOps。核心的靈活基礎架構可讓資料科學家控制使用任何語言、AI架構或運算環境、無論是內部部署或雲端環境、讓他們能夠發揮最佳功能、建置演算法。在任何Kubernetes叢集上、只要使用一個命令、就能輕鬆安裝Cnvrgr-I/O核心。

NetApp ONTAP AI

支援ML和深度學習（DL）工作負載的資料中心參考架構、使用NetApp支援儲存系統、以及搭配Tesla V100 GPU的NVIDIA DGX系統。ONTAP AFFAI採用業界標準的NFS檔案傳輸協定、透過100Gb乙太網路、為客戶提供高效能的ML/DL基礎架構、使用標準資料中心技術來降低實作與管理成本。ONTAP使用標準化的網路和傳輸協定、ONTAP 讓AI能夠整合到混合雲環境、同時維持作業一致性和簡易性。作為預先驗證的基礎架構解決方案、ONTAP Realize AI可縮短部署時間與風險、大幅降低管理成本、讓客戶更快實現價值。

NVIDIA DeepOps

DeepOps是NVIDIA的開放原始碼專案、使用Ansible可根據最佳實務做法、自動部署GPU伺服器叢集。DeepOps是模組化的、可用於各種部署工作。本文件及其所說明的驗證作業中、DeepOps用於部署Kubernetes叢集、其中包含GPU伺服器工作節點。如需詳細資訊、請參閱 "[DeepOps網站](#)"。

NetApp Trident

Trident是NetApp開發與維護的開放原始碼儲存協調工具、可大幅簡化Kubernetes工作負載的持續儲存設備建立、管理與使用。Trident本身就是Kubernetes原生應用程式、直接在Kubernetes叢集內執行。Kubernetes使用者（開發人員、資料科學家、Kubernetes系統管理員等）可以使用他們已經熟悉的標準Kubernetes格式、建立、管理及與持續儲存磁碟區互動。同時、他們也能善用NetApp先進的資料管理功能、以及採用NetApp技術的資料架構。Trident將持續儲存設備的複雜度抽象化、使其易於使用。如需詳細資訊、請參閱 "[Trident網站](#)"。

NetApp StorageGRID

NetApp StorageGRID 功能區是軟體定義的物件儲存平台、可提供簡單、類似雲端的儲存設備、讓使用者使用S3傳輸協定來存取、以滿足這些需求。支援跨網際網路連線站台多個節點的橫向擴充系統、不受距離限制。StorageGRID有了NetApp的智慧型原則引擎StorageGRID、使用者可以選擇跨站台的銷毀編碼物件、在遠端站台之間進行地理恢復或物件複寫、以將WAN存取延遲降至最低。本解決方案提供優異的私有雲主要物件儲存資料庫。StorageGRID

NetApp Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP 的資料管理軟體具備AWS、Google Cloud Platform和Microsoft Azure等公有雲供應商的靈活彈性、可為使用者資料提供控制、保護和效率。NetApp是以NetApp解決方案儲存軟體為基礎打造的雲端原生資料管理軟體、可為使用者提供卓越的通用儲存平台、滿足雲端資料需求。Cloud Volumes ONTAP ONTAP在雲端和內部部署使用相同的儲存軟體、讓使用者能夠享有Data Fabric的價值、而無需訓練IT人員採用全新的方法來管理資料。

對於對混合雲部署模式感興趣的客戶、Cloud Volumes ONTAP 在大多數公有雲中、可提供相同的功能和領先同級的效能、在任何環境中都能提供一致且無縫的使用者體驗。

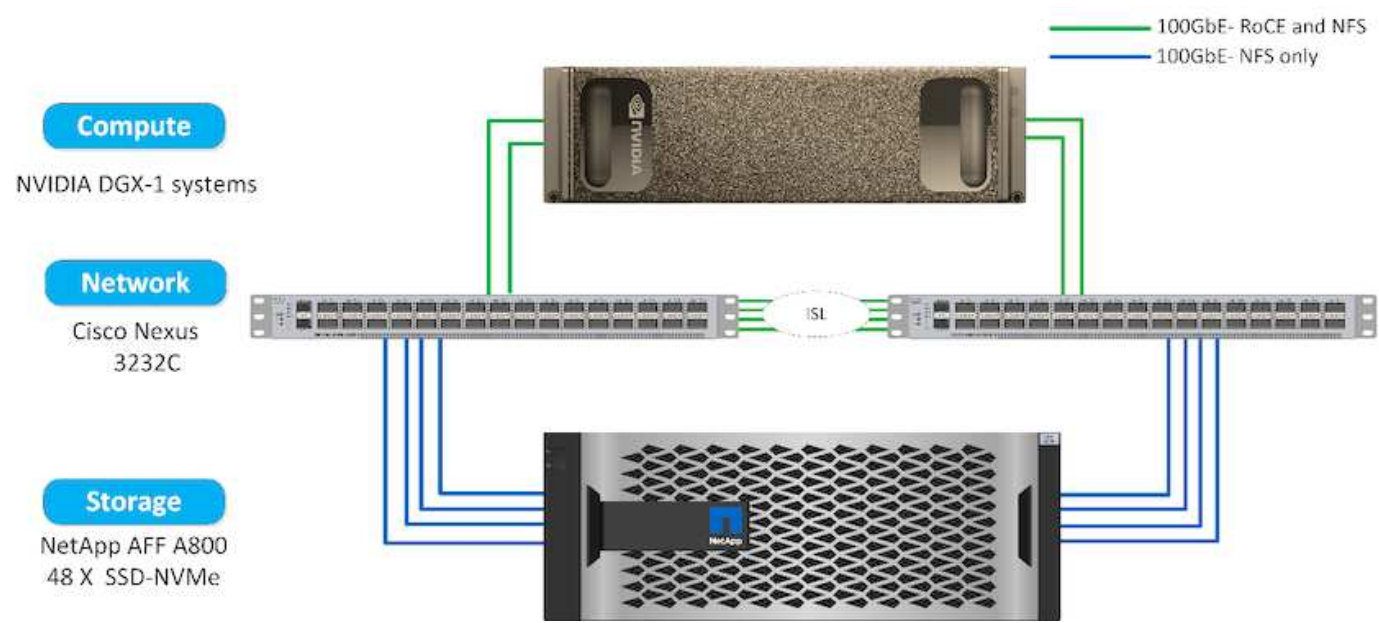
硬體與軟體需求

本節涵蓋ONTAP 有關整個解決方案的技術要求。

硬體需求

雖然硬體需求取決ONTAP 於特定的客戶工作負載、但從單一GPU到機架規模的組態、都能以任何規模部署、以進行資料工程、模型訓練及正式作業提示、以利大規模的ML/DL作業。如需ONTAP 更多關於AI的資訊、請參閱 ["AI網站ONTAP"](#)。

此解決方案已通過DGX-1系統的運算驗證、NetApp AFF 支援A800儲存系統、以及Cisco Nexus 3232C的網路連線能力驗證。本驗證所使用的功能豐富、最多可支援10個DGX-1系統、以滿足大多數ML/DL工作負載的需求。AFF下圖顯示ONTAP 此驗證中用於模型訓練的AI解決方案。



為了將此解決方案延伸至公有雲、Cloud Volumes ONTAP 可將其與雲端GPU運算資源一起部署、並整合至混合雲資料架構中、讓客戶能夠使用適合任何特定工作負載的任何資源。

軟體需求

下表顯示本解決方案驗證所使用的特定軟體版本。

元件	版本
Ubuntu	18.04.4 LTS
NVIDIA DGX OS	4.4.0
NVIDIA DeepOps	20.02.1
Kubernetes	1.15
掌舵	3.1.0
cnvrg-io	3.0.00.0
NetApp ONTAP	9.6P4

在本解決方案驗證中、Kubernetes已部署為DGX-1系統上的單節點叢集。對於大規模部署、應部署獨立的Kubernetes主節點、以提供高可用度的管理服務、並為ML和DL工作負載保留寶貴的DGX資源。

解決方案部署與驗證詳細資料

下列各節將討論解決方案部署與驗證的詳細資料。

支援AI部署ONTAP

部署AI需要安裝和組態網路、運算和儲存硬體。ONTAP關於部署AI基礎架構的具體指示ONTAP 不在本文的討論範圍之內。如需詳細的部署資訊、請參閱 ["NVA-1121-Deploy：採用ONTAP NVIDIA技術的NetApp支援"](#)。

針對此解決方案驗證、已建立單一磁碟區並掛載至DGX-1系統。然後將該掛載點掛載到容器中、以便進行訓練時存取資料。對於大規模部署、NetApp Trident會自動建立及安裝磁碟區、以免除管理成本、並讓終端使用者能夠管理資源。

Kubernetes部署

若要使用NVIDIA DeepOps部署及設定Kubernetes叢集、請從部署跳接主機執行下列工作：

1. 依照上的指示下載NVIDIA DeepOps ["入門頁面"](#) 在NVIDIA DeepOps GitHub網站上。
2. 依照上的指示、在叢集中部署Kubernetes ["Kubernetes部署指南"](#) 在NVIDIA DeepOps GitHub網站上。



若要讓DeepOps Kubernetes部署正常運作、所有Kubernetes主節點和工作節點上都必須有相同的使用者。

如果部署失敗、請在「depops/config/group_vars/k8s-cluster.yml」中、將「kubectll_localhost」的值變更為「假」、然後重複步驟2。只有當值為「kubectll_localhost」時、「Copy kubectll二進位到Ansible host」工作才會執行、這項工作仰賴已知記憶體使用問題的擷取Ansible模組。這些記憶體使用量問題有時可能導致工作失敗。如果工作因為記憶體問題而失敗、則部署作業的其餘部分將無法成功完成。

如果在您將「kubectll_localhost」的值變更為「假」之後、成功完成部署、則必須手動將「kubectll二進位」從Kubernetes主節點複製到部署跳接主機。您可以在特定主節點上直接執行「that kubectll」命令、找到「kubectll二進位」的位置。

Cnvrg-IO部署

使用Helm部署cnvrg核心

使用任何叢集、內部部署、Minikube,或任何雲端叢集（例如、KS、EKS和GKE）、Helm是快速部署cnvrg的最簡單方法。本節說明如何在安裝Kubernetes的內部部署（DGX-1）執行個體上安裝cnvrg。

先決條件

在完成安裝之前、您必須先在本機機器上安裝並準備下列相依項目：

- Kubectll
- helm 3.x
- Kubernetes叢集1.15以上

使用Helm進行部署

1. 若要下載最新的cnvrg helm圖表、請執行下列命令：

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

2. 部署cnvrg之前、您需要叢集的外部IP位址、以及要部署cnvrg的節點名稱。若要在內部部署的Kubernetes叢集上部署cnvrg、請執行下列命令：

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

3. 執行「helm install」命令。所有服務和系統都會自動安裝在叢集上。此程序最多可能需要15分鐘。
4. 「helm install」命令最多可能需要10分鐘。部署完成後、請前往新部署的cnvrg的URL、或將新叢集新增為組織內部的資源。「helm」命令會通知您正確的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. 當所有容器的狀態都在執行或完成時、表示已成功部署cnvrg。其外觀應類似於下列輸出範例：

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-2zbp	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-kcvt9	1/1	Running	0	2m
seeder-wh655	0/1	Completed	0	2m
speaker-5sghr	1/1	Running	0	2m

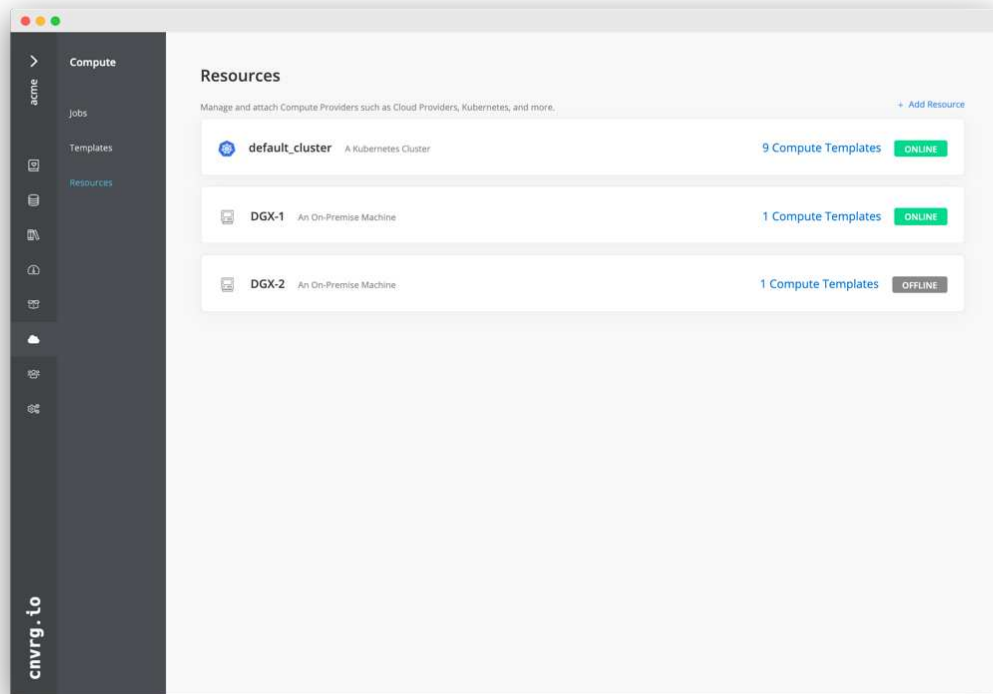
ResNet50和Chest X射線資料集的電腦願景模型訓練

Cnvrg-IO AI OS部署在Kubernetes設定上ONTAP、採用NVIDIA DGX系統的NetApp AI架構上。為了進行驗證、我們使用NIH Chest X光資料集、其中包含已取消識別的胸前X光影像。影像採用的是PNG格式。資料由NIH臨床中心提供、可透過取得 ["NIH下載網站"](#)。我們使用250 GB的資料樣本、在15個類別中使用627、615個影像。

此資料集已上傳至cnvrg平台、並從NetApp AFF S16A800儲存系統快取至NFS匯出。

設定運算資源

Cnvrg架構和中繼排程功能可讓工程師和IT專業人員將不同的運算資源附加至單一平台。在我們的設定中、我們使用的叢集cnvrg與執行深度學習工作負載所部署的叢集cnvrg相同。如果您需要附加其他叢集、請使用GUI、如下面的快照所示。

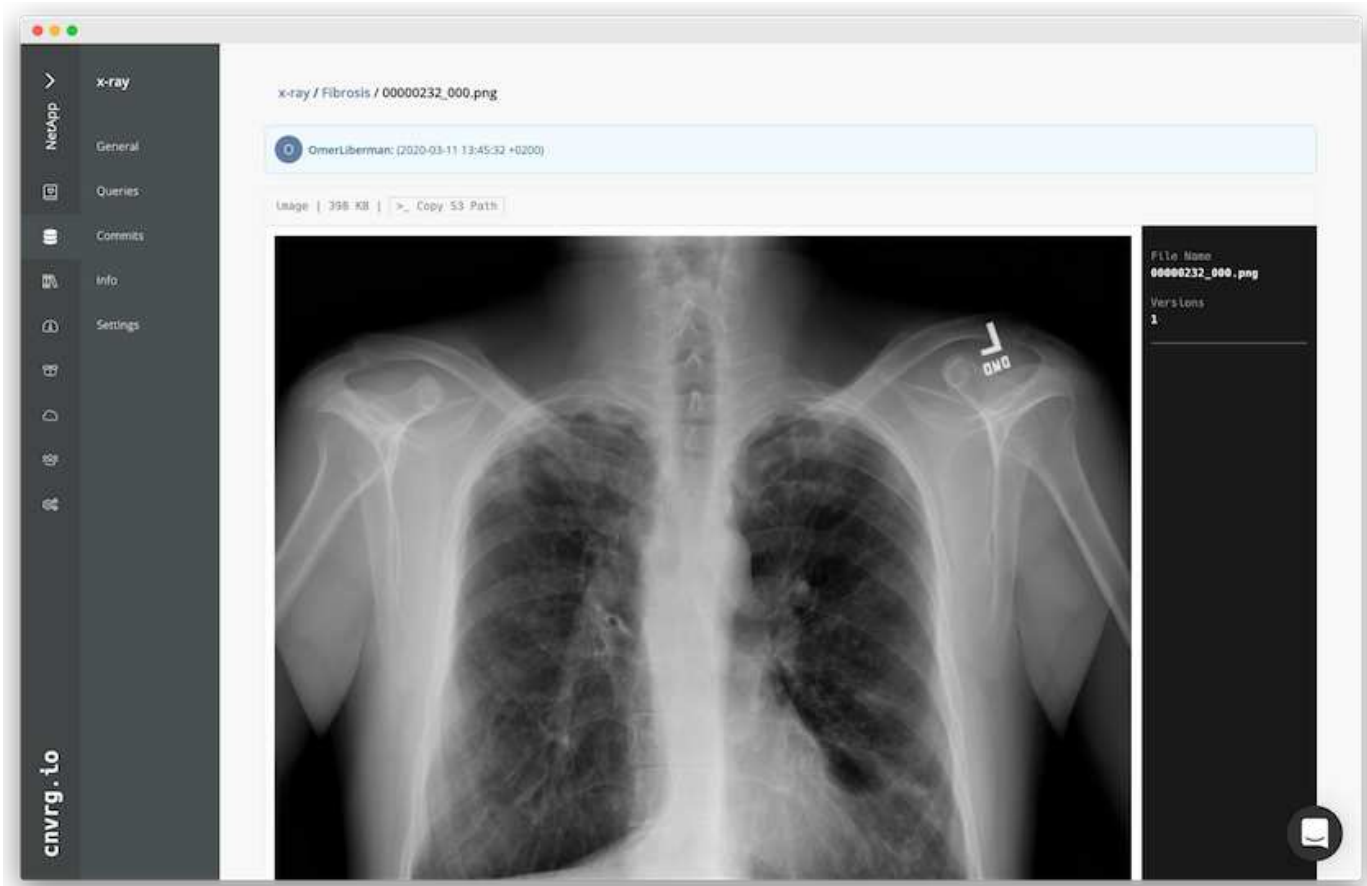


負載資料

若要將資料上傳至cnvrg平台、您可以使用GUI或cnvrg CLI。對於大型資料集、NetApp建議使用CLI、因為它是強大、可擴充且可靠的工具、可處理大量檔案。

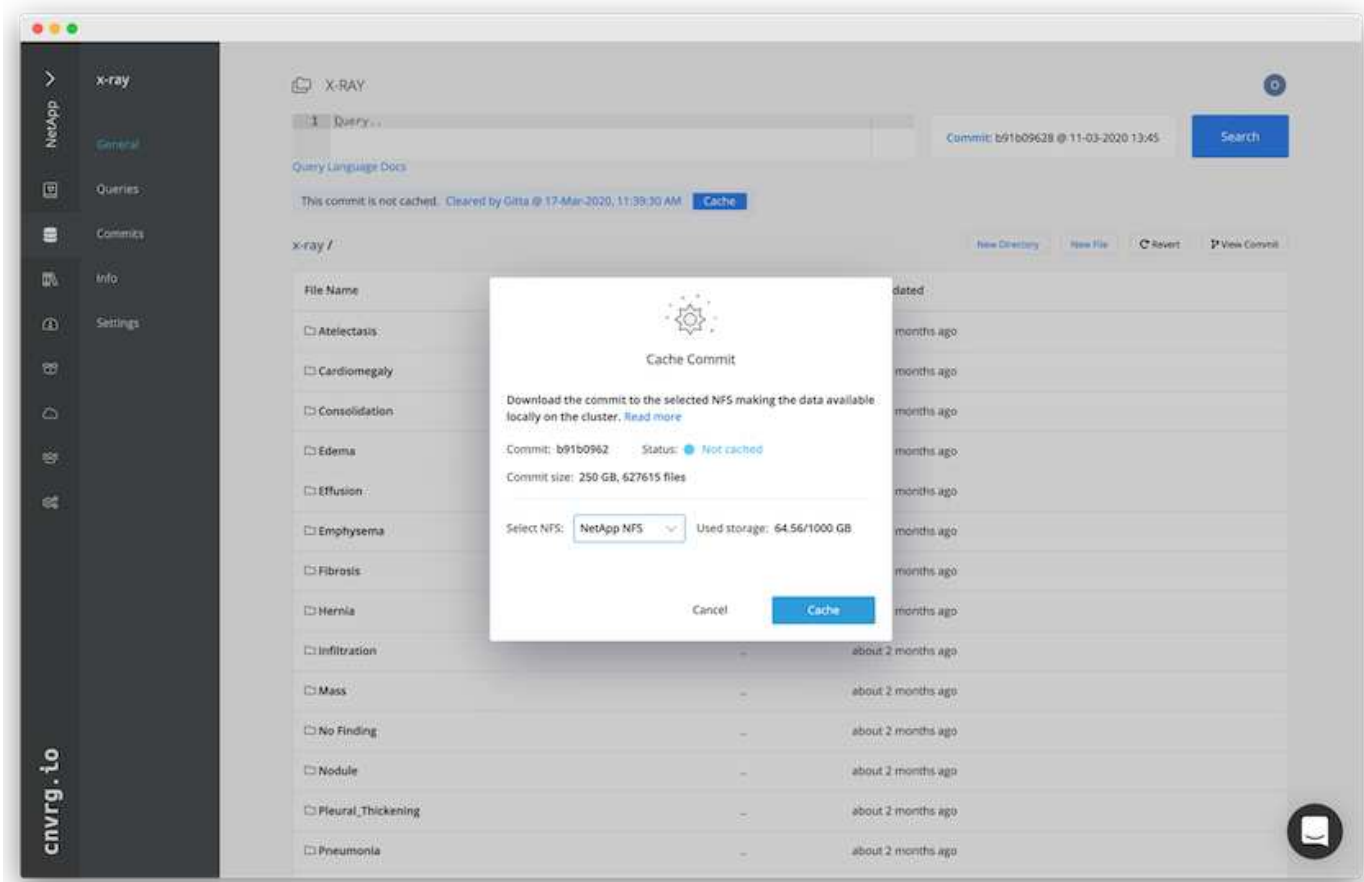
若要上傳資料、請完成下列步驟：

1. 下載 "Cnvrg CLI"。
2. 瀏覽至X光目錄。
3. 使用「cnvrg data init」命令、在平台中初始化資料集。
4. 使用「cnvrg data sync」命令、將目錄的所有內容上傳至中央資料湖。資料上傳至中央物件存放區StorageGRID（例如、S3或其他）之後、您就可以使用GUI瀏覽。下圖顯示已載入的胸前X光纖維化影像PNG檔案。此外、cnvrg會將資料版本轉換成資料版本、以便您建置的任何模型都能複製到資料版本。



Cach資料

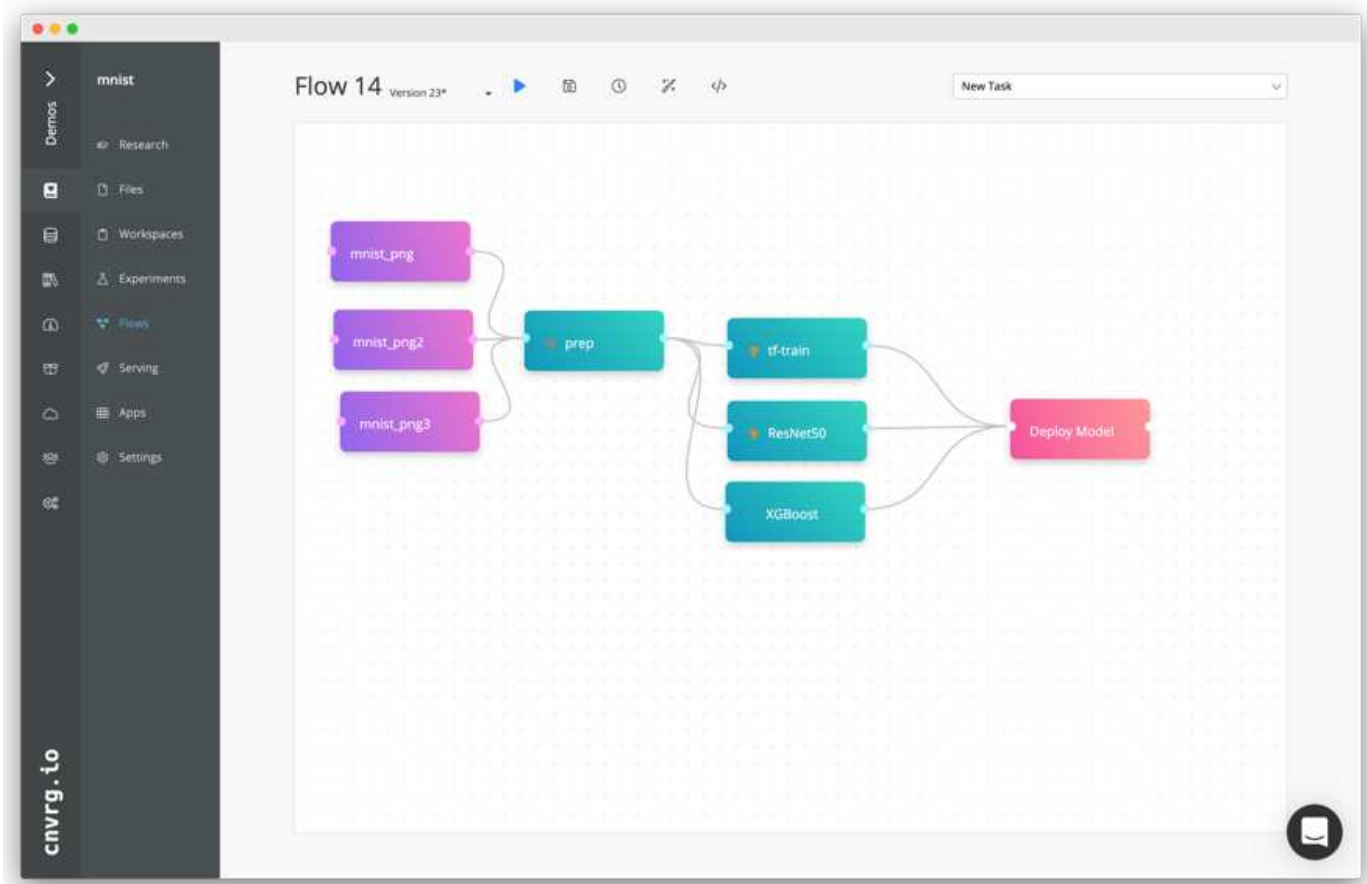
為了加快訓練速度、避免為每個模型訓練和實驗下載超過60萬個檔案、我們在資料一開始上傳至中央資料湖物件存放區之後、就使用了資料快取功能。



使用者按一下「Cache（快取）」後、cnvrg會從遠端物件存放區下載其特定提交的資料、並將其快取至ONTAP「SflexNFS Volume（更新資料）」。資料完成後、即可立即接受訓練。此外、如果資料未使用數天（例如模型訓練或探索）、cnvrg會自動清除快取。

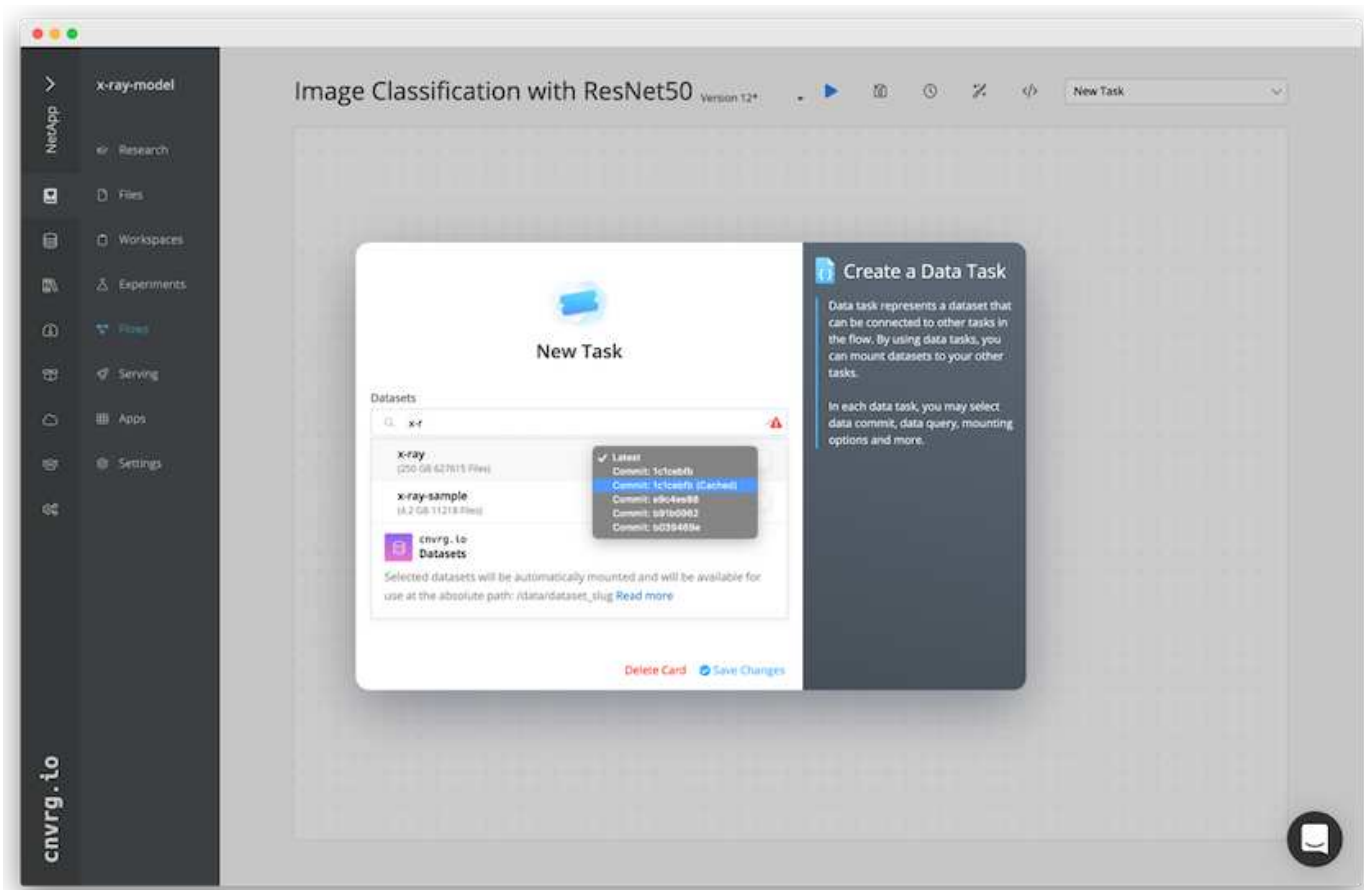
利用快取資料建立**ML**管道

Cnvrg流程可讓您輕鬆建立正式作業ML管線。流程很靈活、可用於任何類型的ML使用案例、並可透過GUI或程式碼建立。流程中的每個元件都能以不同的Docker映像檔在不同的運算資源上執行、因此能夠建置混合雲和最佳化的ML管線。



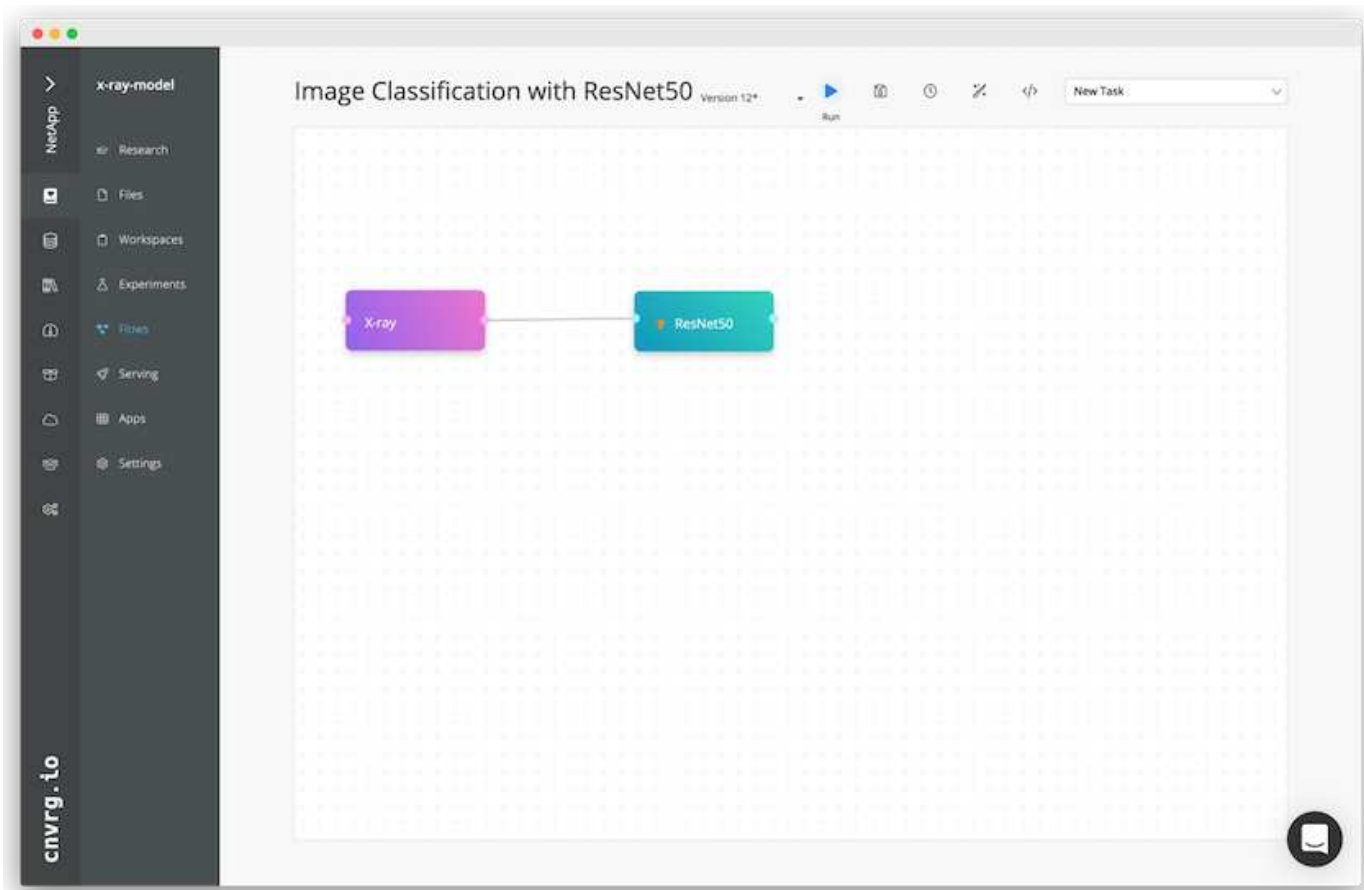
建立胸前X光流程：設定資料

我們將資料集新增至新建立的流程。新增資料集時、您可以選取特定版本（提交）、並指出是否要使用快取版本。在此範例中、我們選取了快取的commit。



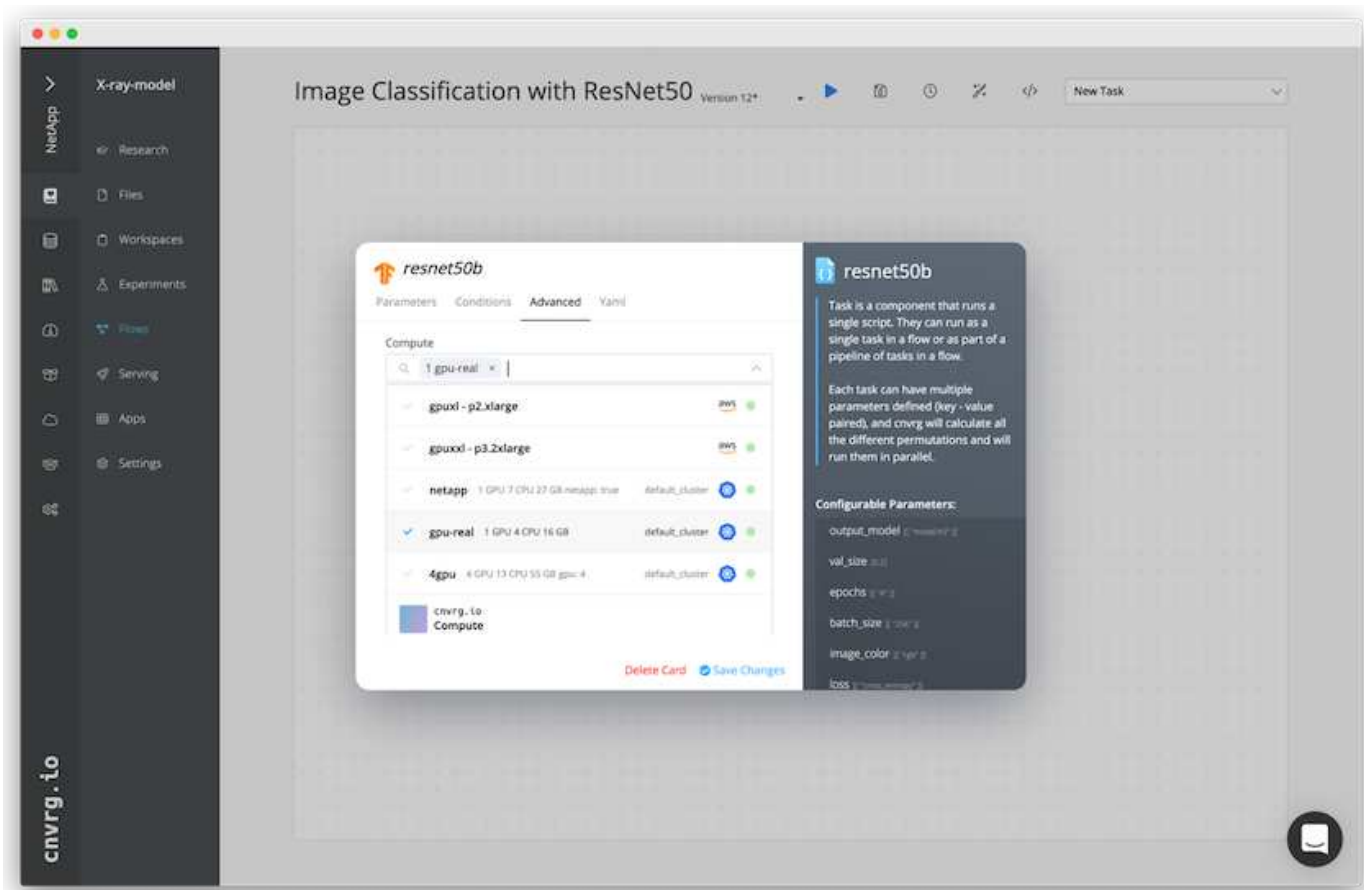
建立胸前X光流程：設定訓練模式：ResNet50

在管道中、您可以新增任何類型的自訂程式碼。在cnvrg中、還有AI程式庫、可重複使用的ML元件集合。AI程式庫中有演算法、指令碼、資料來源及其他解決方案、可用於任何ML或深度學習流程。在此範例中、我們選擇了預先建置的ResNet50模組。我們使用預設參數、例如batch_Size:128、epochs：10等。這些參數可在AI程式庫文件中檢視。下列螢幕快照顯示新流程、其中X光資料集已連線至ResNet50。



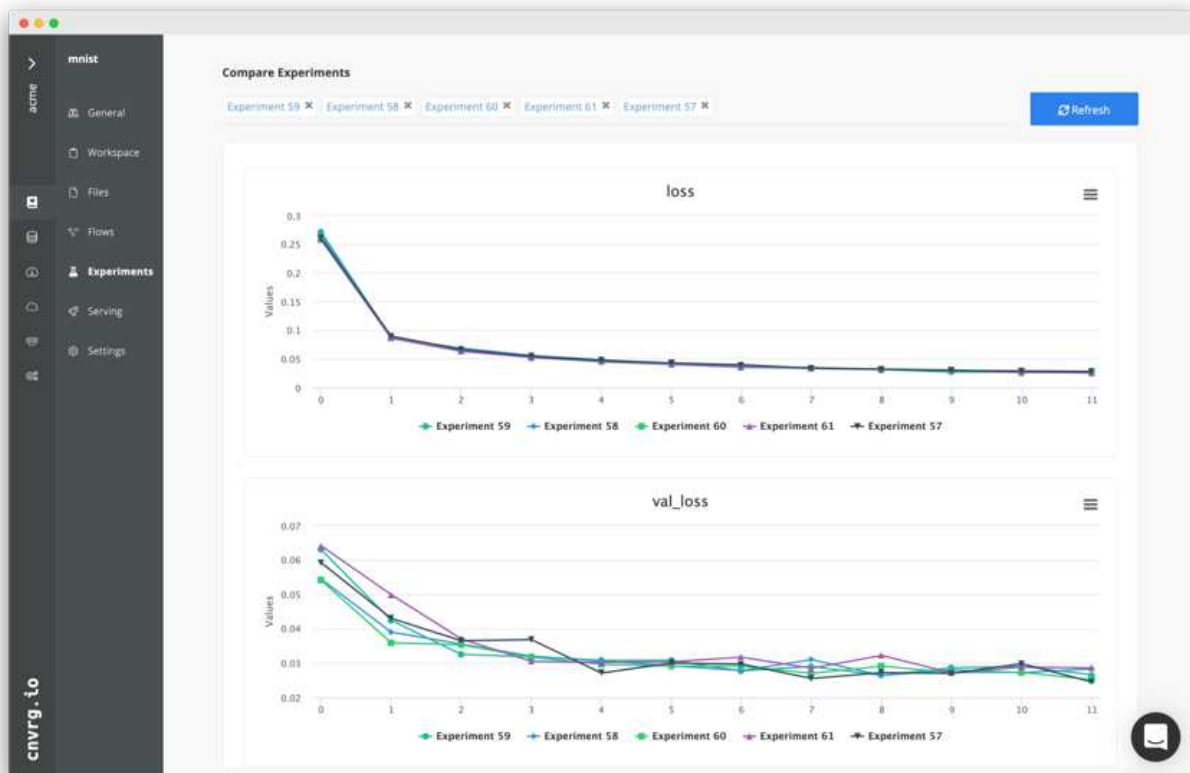
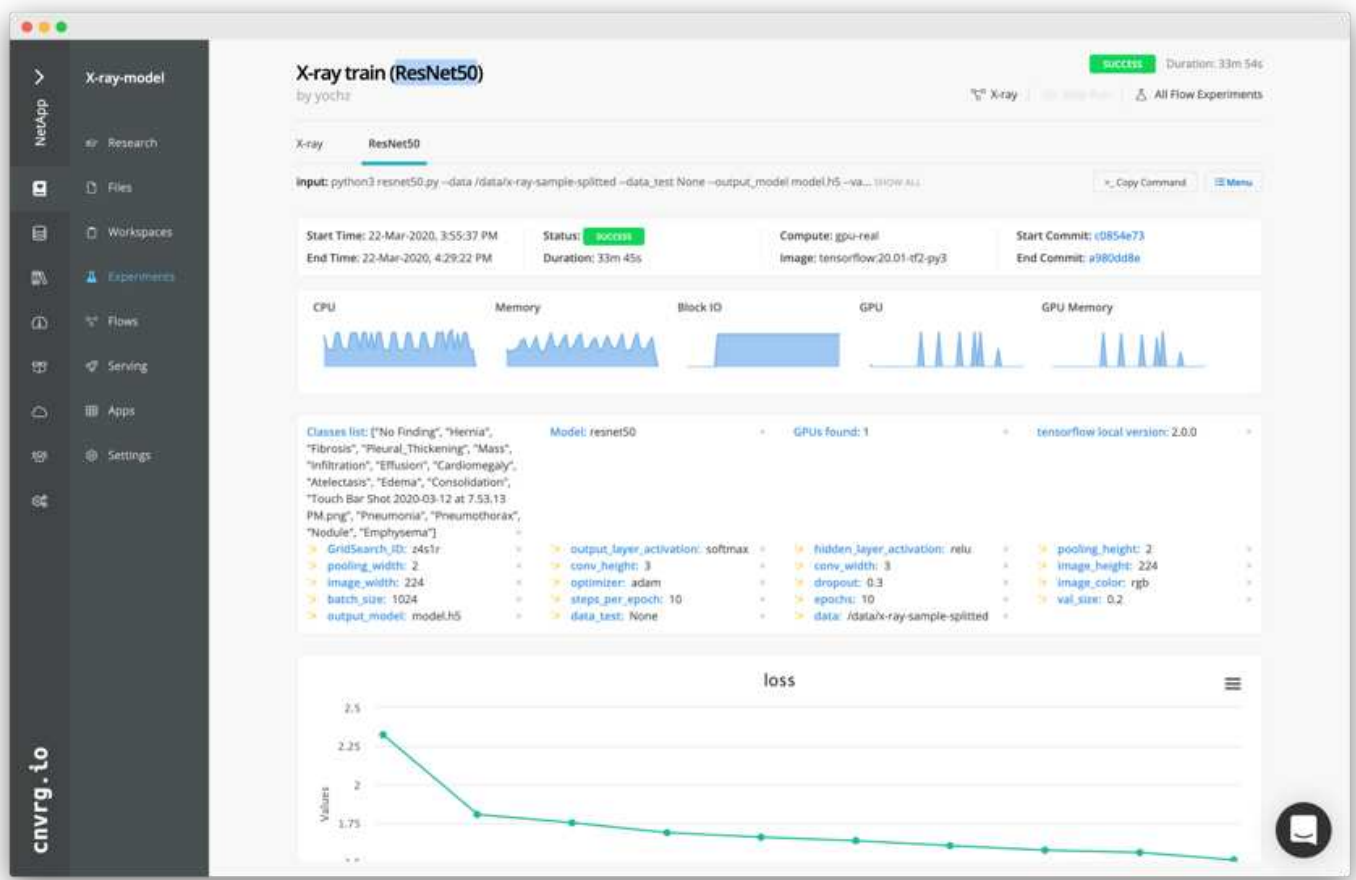
定義ResNet50的運算資源

cnvrg流程中的每個演算法或元件都可以在不同的運算執行個體上執行、並使用不同的Docker映像檔。在我們的設定中、我們想要在採用NetApp ONTAP AI架構的NVIDIA DGX系統上執行訓練演算法。在下圖中、我們選擇了「GPU實際」、這是內部部署叢集的運算範本和規格。我們也建立了範本佇列、並選取多個範本。如此一來、如果無法分配「GPU實際」資源（例如其他資料科學家正在使用該資源）、您就可以新增雲端供應商範本來啟用自動雲端資源爆增功能。下列螢幕快照顯示如何使用GPU Real做為ResNet50的運算節點。



追蹤及監控結果

執行流程之後、cnvrg會觸發追蹤與監控引擎。每次流程執行都會自動記錄並即時更新。超參數、度量、資源使用率（GPU使用率等）、程式碼版本、成品、記錄、「實驗」區段會自動提供這些功能、如下圖所示。



結論

NetApp與cnvrg-IO合作、為客戶提供完整的資料管理解決方案、以利ML與DL軟體開發。支援各種規模的作業、均可提供高效能的運算與儲存、而cnvrg-IO軟體則可簡化資料科學工作流程、並改善資源使用率。ONTAP

感謝

- Mike Oglesby、NetApp技術行銷工程師
- NetApp資深技術總監Santosh Rao

何處可找到其他資訊

若要深入瞭解本文所述資訊、請參閱下列資源：

- Cnvrg-IO ("<https://cnvrg.io>") :
 - Cnvrg核心 (免費ML平台)
<https://cnvrg.io/platform/core>
 - Cnvrg文件
["https://app.cnvrg.io/docs"](https://app.cnvrg.io/docs)
- NVIDIA DGX-1伺服器：
 - NVIDIA DGX-1伺服器
<https://www.nvidia.com/en-us/data-center/dgx-1/>
 - NVIDIA Tesla V100 Tensor Core GPU
<https://www.nvidia.com/en-us/data-center/tesla-v100/>
 - NVIDIA GPU雲端 (NGC)
<https://www.nvidia.com/en-us/gpu-cloud/>
- NetApp AFF 系統：
 - 資料表AFF
<https://www.netapp.com/us/media/d-3582.pdf>
 - NetApp FlashAdvantage for AFF 功能
<https://www.netapp.com/us/media/ds-3733.pdf>
 - 2.x文件ONTAP
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>

- NetApp FlexGroup 技術報告

<https://www.netapp.com/us/media/tr-4557.pdf>

- 適用於容器的NetApp持續儲存設備：

- NetApp Trident

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

- NetApp互通性對照表：

- NetApp 互通性對照表工具

<http://support.netapp.com/matrix>

- AI網路：ONTAP

- Cisco Nexus 3232C交換器

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

- Mellanox Spectrum 2000系列交換器

http://www.mellanox.com/page/products_dyn?product_family=251&mtag=sn2000

- ML架構與工具：

- 達利

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

- TensorFlow：適用於所有人的開放原始碼機器學習架構

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

- Horovod：Uber的開放原始碼分散式深度學習架構、適用於TensorFlow

<https://eng.uber.com/horovod/>

- 在Container執行時間生態系統中啟用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筆記型電腦伺服器

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

- 資料集與基準測試：

- NIH胸前X光資料集

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

- 王小鬆、彭葉文、盧、陸志勇、MohammadhADI Bagheri、Ronald Summers、ChestX-RAY 8：醫院規模的ChestX-Ray X光資料庫、以及一般胸病弱監督分類與本地化的基準測試、IEEE CVPR、第頁3462-3471、2017TR-4841-0620

AI在Edge大會上推斷- NetApp採用Lenovo ThinkSystem -解決方案設計

TR-4886：Edge的AI推斷-採用Lenovo ThinkSystem的NetApp解決方案設計

Sathish Thyagarajan、NetApp Miroslav Hodak、聯想

摘要

許多新興的應用程式案例（例如進階驅動程式輔助系統（ADAS）、Industry 4.0、智慧城市和物聯網（IoT））、都需要在接近零延遲的情況下處理持續的資料流。本文件說明運算與儲存架構、可在符合這些需求的邊緣環境中、於NetApp儲存控制器和Lenovo ThinkSystem伺服器上部署GPU型人工智慧（AI）提示。本文件也提供業界標準MLPerf推斷基準測試的效能資料、以評估配備NVIDIA T4 GPU之邊緣伺服器上的各種推斷工作。我們會調查離線、單一串流和多重串流推斷案例的效能、並顯示採用具成本效益的共享網路儲存系統的架構具有高效能、可為多個邊緣伺服器提供資料和模型管理的集中點。

簡介

公司越來越多地在網路邊緣產生大量資料。為了從智慧型感應器和IoT資料中獲得最大價值、企業組織正在尋求能夠實現邊緣運算的即時事件串流解決方案。因此、運算需求嚴苛的工作會越來越多地在資料中心以外的邊緣執行。AI推斷是這種趨勢的驅動因素之一。邊緣伺服器可為這些工作負載提供足夠的運算能力、尤其是在使用加速器時、但有限的儲存設備通常是個問題、尤其是在多伺服器環境中。在本文件中、我們將說明如何在邊緣環境中部署共享儲存系統、以及它如何在不影響效能的情況下、為AI推斷工作負載帶來好處。

本文件說明邊緣AI推斷的參考架構。它結合多部Lenovo ThinkSystem邊緣伺服器與NetApp儲存系統、打造易於部署和管理的解決方案。本指南旨在作為在各種情況下實際部署的基礎指南、例如裝有多部攝影機和工業感測器的工廠場地、零售交易中的銷售點（POS）系統、或是能識別自主車輛視覺異常狀況的全自駕（FSD）系統。

本文件涵蓋由Lenovo ThinkSystem SE350 Edge Server和入門級NetApp AFF 支援中心與EF系列儲存系統所組成之運算與儲存組態的測試與驗證。參考架構為AI部署提供高效率且具成本效益的解決方案、同時也提供全方位的資料服務、整合式資料保護、無縫擴充性、以及採用NetApp ONTAP 功能與NetApp SANtricity 功能的NetApp資料管理軟體所提供的雲端連線資料儲存設備。

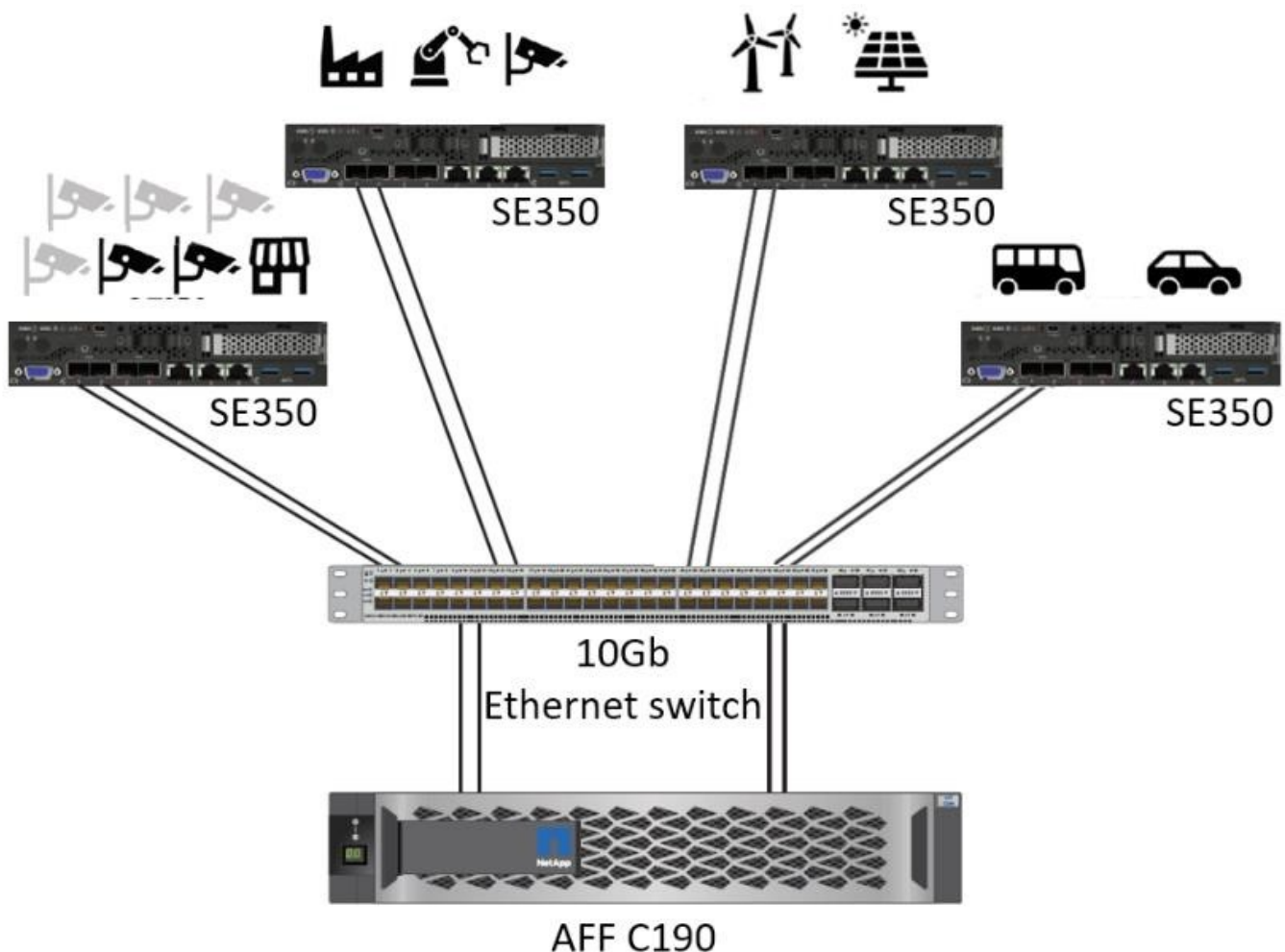
目標對象

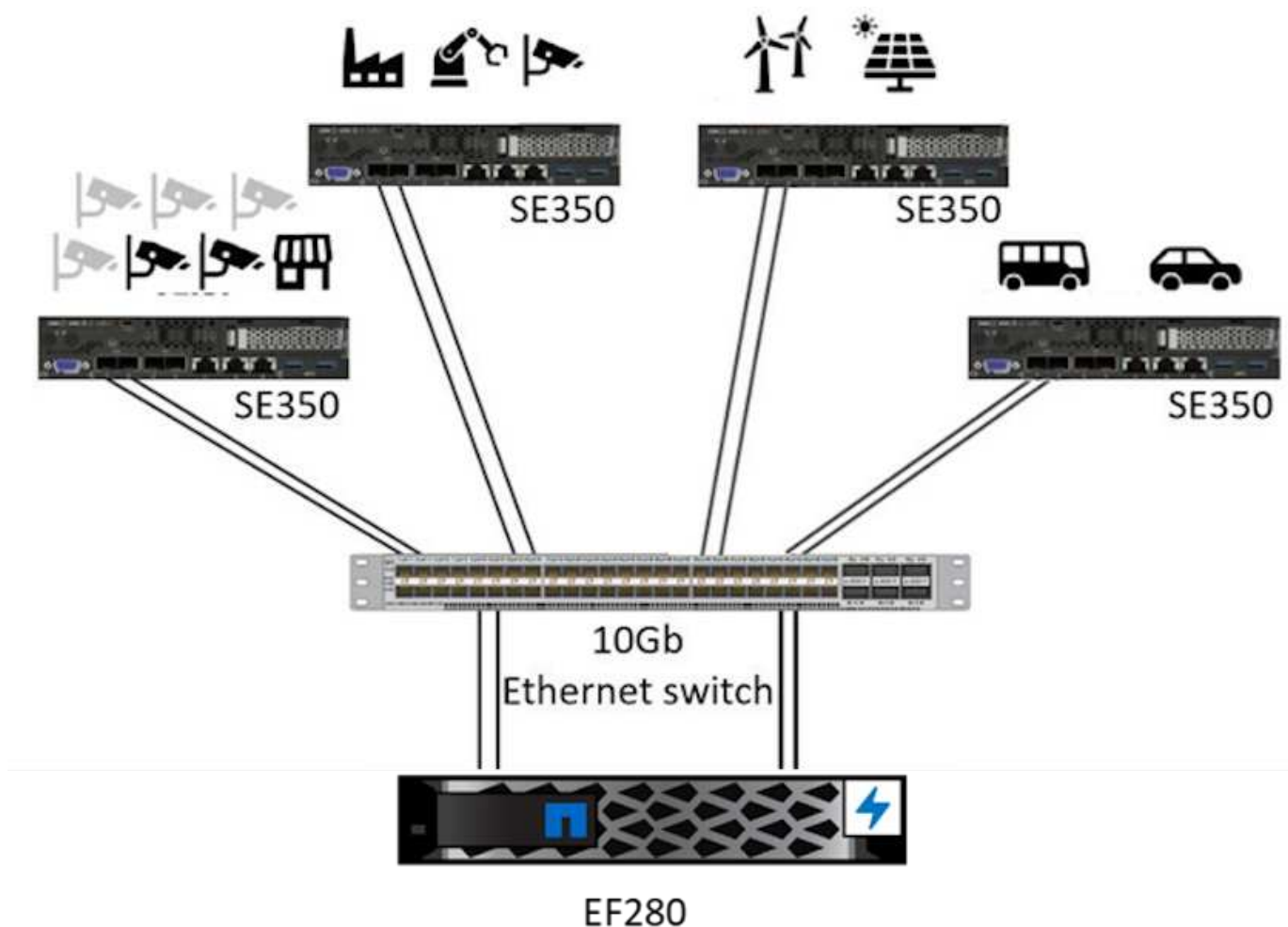
本文適用於下列對象：

- 想要將邊緣AI產品化的企業領導者和企業架構設計師。
- 資料科學家、資料工程師、AI /機器學習（ML）研究人員、以及AI系統開發人員。
- 設計解決方案以開發AI/ML模型和應用程式的企業架構設計師。
- 資料科學家和AI工程師正在尋求有效方法來部署深度學習（DL）和ML模型。
- 負責部署及管理Edge推斷模型的Edge裝置管理員和Edge Server管理員。

解決方案架構

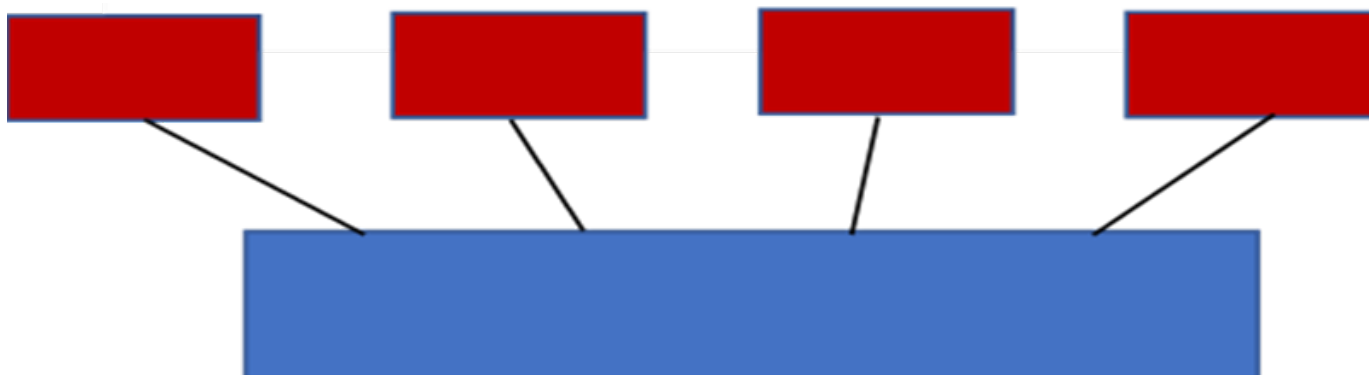
這款Lenovo ThinkSystem伺服ONTAP 器和NetApp的可靠性或NetApp SANtricity 的不二之選儲存解決方案、是專為處理大型資料集的AI推斷、使用GPU的處理能力、以及傳統的CPU。這項驗證可展現高效能與最佳資料管理、其架構使用單一或多部Lenovo SR350邊緣伺服器、並與單AFF 一NetApp效益儲存系統互連、如下圖所示。





下圖中的邏輯架構總覽顯示此架構中運算和儲存元素的角色。具體而言、它顯示下列內容：

- 邊緣運算裝置會對從攝影機、感應器等接收的資料執行推斷。
- 一種共享儲存元件、可滿足多種用途：
 - 為執行推斷所需的推斷模型及其他資料提供集中位置。運算伺服器可直接存取儲存設備、並在整個網路上使用推斷模型、而不需要在本機複製。
 - 更新的機型將在此推播。
 - 歸檔邊緣伺服器接收的輸入資料、以供日後分析。例如、如果邊緣裝置連接至攝影機、儲存元件會保留攝影機拍攝的影片。



紅色	藍色
聯想運算系統	NetApp AFF 產品儲存系統
邊緣裝置會對來自攝影機、感應器等輸入執行推斷。	共享儲存設備、保留來自邊緣裝置的推斷模型和資料、以供日後分析。

此NetApp與Lenovo解決方案提供下列主要效益：

- GPU加速邊緣運算。
- 部署多部邊緣伺服器、從共享儲存設備進行備份與管理。
- 強大的資料保護功能、可滿足低恢復點目標（RPO）和恢復時間目標（RTO）、而且不會遺失資料。
- 利用NetApp Snapshot複本與複製技術來最佳化資料管理、以簡化開發工作流程。

如何使用此架構

本文件驗證所建議架構的設計與效能。不過、我們尚未測試某些軟體層級的元件、例如容器、工作負載、或模型管理、以及與內部部署的雲端或資料中心的資料同步、因為它們是針對部署案例而設計。這裡有多種選擇。

在容器管理層級、Kubernetes Container管理是一個不錯的選擇、無論是完整上游版本（Canonical）或是適合企業部署的修改版本（Red Hat）、都能獲得良好支援。。"[NetApp AI Control Plane](#)" 使用NetApp Trident和新增的 "[NetApp DataOps工具套件](#)" 提供內建的可追蹤性、資料管理功能、介面和工具、讓資料科學家和資料工程師能夠與NetApp儲存設備整合。Kubernetes的ML工具套件Kubeflow提供額外的AI功能、並支援TensorFlow服務或NVIDIA Triton Inference伺服器等多種平台上的模型版本設定和KFServing。另一個選項是NVIDIA EGX平台、可提供工作負載管理功能、並可存取支援GPU的AI推斷容器目錄。不過、這些選項可能需要投入大量心力和專業知識才能投入生產、而且可能需要協力廠商獨立軟體廠商（ISV）或顧問的協助。

解決方案領域

AI推斷和邊緣運算的主要優點在於、裝置能夠以高品質、無延遲的方式來運算、處理及分析資料。本文件中有太多邊緣運算使用案例可說明、但以下是幾個主要案例：

汽車：自主車輛

典型的邊緣運算圖示為自動車輛（AV）的進階駕駛輔助系統（ADAS）。無磁碟汽車的AI必須迅速處理來自攝影機和感應器的大量資料、才能成為成功的安全駕駛者。在物件與人之間解讀太久可能意味著生命或死亡、因此盡可能將資料處理在車輛附近是非常重要的。在此情況下、一或多個邊緣運算伺服器會處理來自攝影機、雷達、LIDAR及其他感應器的輸入、而共享儲存設備則會保留推斷模型、並儲存來自感應器的輸入資料。

醫療：病患監控

AI和邊緣運算最大的影響之一、就是能夠強化對在家中照護和深切治療單位（ICU）長期病患的持續監控能力。來自監測胰島素水平、呼吸、神經活動、心臟心律和腸胃功能的邊緣裝置的資料、需要即時分析資料、但必須立即採取行動、因為行動的時間有限、無法拯救某人的生命。

零售：無收銀員付款

邊緣運算可以推動AI和ML、協助零售商縮短結帳時間、並增加腳步流量。無收銀機系統支援各種元件、例如：

- 驗證與存取：將實體購物者連線至已驗證的帳戶、並允許存取零售空間。
- 庫存監控：使用感測器、RFID標籤和電腦視覺系統、協助確認購物者選擇或取消選擇商品。

在這裡、每個邊緣伺服器都會處理每個結帳櫃位、而共享儲存系統則是中央同步點。

金融服務：資訊站的人類安全與防範詐騙

銀行組織正使用AI和邊緣運算技術來創新及創造個人化的銀行體驗。互動式資訊站使用即時資料分析和AI推斷功能、現在可讓ATM不僅協助客戶提領資金、還能透過從攝影機擷取的影像主動監控資訊站、以識別人類安全或詐騙行為的風險。在此案例中、邊緣運算伺服器和共享儲存系統會連線至互動式多媒體資訊站和攝影機、以AI推斷模式協助銀行收集和處理資料。

製造業：產業4.0

第四次產業革命（產業4.0）已經開始、也伴隨著智慧工廠和3D列印等新興趨勢。為了準備迎接資料導向的未來、我們整合了大型機器對機器（M2M）通訊與IoT、無需人為介入、即可提升自動化程度。製造業已高度自動化、加上AI功能、自然是長期趨勢的延續。AI可實現自動化作業、並藉由電腦願景和其他AI功能來實現自動化。您可以自動化品質控制、或是仰賴人的願景或決策來執行更快速分析工廠內組裝線路上的材料、以協助製造廠符合所需的ISO安全與品質管理標準。在此處、每部運算邊緣伺服器都會連接到監控製造流程的感測器陣列、並視需要將更新的推斷模型推送至共享儲存設備。

電信：除鏽偵測、塔式檢查及網路最佳化

電信產業使用電腦願景和AI技術來處理影像、這些影像會自動偵測出各種生鏽、並識別含有侵蝕的電池塔、因此需要進一步檢查。近年來、利用無人機影像和AI模型來識別塔內的不同區域、以分析鐵鏽、表面碎裂和侵蝕。AI技術的需求持續成長、可有效檢查電信基礎架構和電池塔、定期評估是否有老化問題、並在需要時立即修復。

此外、電信領域的另一項新興使用案例是使用AI和ML演算法來預測資料流量模式、偵測具備5G功能的裝置、以及自動化及強化多重輸入和多重輸出（MIMU）能源管理。在無線塔上使用的是多重可擴充網路容量的MIMO-硬體、但也會增加能源成本。部署於行動站台的「MIMO睡眠模式」有多種ML模式、可預測無線電的有效使用、並有助於降低行動網路營運者（MNO）的能源消耗成本。AI推斷與邊緣運算解決方案可協助MNO減少資料中心來回傳輸的資料量、降低TCO、最佳化網路作業、並改善終端使用者的整體效能。

技術總覽

本節說明此AI解決方案的技術基礎。

NetApp AFF 產品

最先進的NetApp AFF®不中斷儲存系統可讓AI在邊緣部署、以領先業界的效能、優異的靈活性、雲端整合及同級最佳的資料管理功能、滿足企業儲存需求。專為Flash所設計的NetApp AFF 支援功能、有助於加速、管理及保護業務關鍵資料。

- 入門級NetApp AFF 支援儲存系統是以FAS2750硬體和SSD Flash媒體為基礎
- HA組態中有兩個控制器



NetApp入門級AFF 的C190儲存系統支援下列功能：

- 磁碟機數量上限為24 x 960GB SSD
- 兩種可能的組態：
 - 乙太網路（10GbE）：4個10GBASE-T（RJ-45）連接埠
 - 統一化（16GB FC或10GbE）：4個統一化目標介面卡2（UTA2）連接埠
- 最大有效容量為50.5 TB



對於NAS工作負載、單一入門級AFF 的432C190系統可支援4.4GBps的連續讀取處理量、以及230K IOPS、以1毫秒或更短的延遲時間進行小型隨機讀取。

NetApp AFF 解決方案-A220

NetApp也提供其他入門級儲存系統、為大規模部署提供更高的效能與擴充性。對於NAS工作負載、單一入門級AFF 的AS20220系統支援：

- 連續讀取的處理量為6.2GBps
- 375K IOPS可在延遲1毫秒或更短時間內進行小型隨機讀取
- 最大磁碟機數為144x 960GB、3.8TB或7.6TB SSD
- 可用容量可擴充至大於1PB的部分AFF

NetApp AFF 產品-A250

- 最大有效容量為35PB、最大橫向擴充為2-24個節點（12個HA配對）
- 效能比AFF 不支援的A220高出≥45%
- 440k IOPS隨機讀取@1ms
- 以最新的NetApp ONTAP 產品資訊版本ONTAP 為基礎：《NetApp產品資訊
- 利用兩個25GB乙太網路進行HA和叢集互連

NetApp E系列EF系統

EF系列是入門級與中階All Flash SAN儲存陣列系列產品、可加速資料存取、並利用NetApp SANtricity 的Efi2軟體、更快從其中獲取價值。這些系統同時提供SAS和NVMe Flash儲存設備、並提供經濟實惠的極致IOPS、100微秒內的回應時間、以及高達44GBps的頻寬、是混合式工作負載和嚴苛應用程式（例如AI推斷和高效能運算（HPC）的理想選擇。

下圖顯示NetApp EF280儲存系統。



NetApp EF280

- 32GB/16GB FC、25GB/10Gb iSCSI及12Gb SAS支援
- 最大有效容量為96個磁碟機、總計1.5PB
- 10Gbps的處理量（循序讀取）
- 300K IOPs（隨機讀取）
- NetApp EF280是NetApp產品組合中成本最低的All Flash Array（AFF）

NetApp EF300

- 24個NVMe SSD磁碟機、總容量為367TB
- 擴充選項總計240x NL-SAS HDD、96x SAS SSD或組合
- 100GB NVMe / IB、NVMe / RoCE、iSER / IB和SRP/IB
- 32GB NVMe/FC、FCP
- 25GB iSCSI
- 20Gbps（循序讀取）
- 670K IOPs（隨機讀取）



如需詳細資訊、請參閱 ["NetApp EF系列NetApp EF系列All Flash Array EF600、F300、EF570及EF280資料表"](#)。

NetApp ONTAP 產品9.

NetApp最新一代的儲存管理軟體《支援》（NetApp）版本、可讓企業將基礎架構現代化、並移轉至雲端就緒的

資料中心。ONTAP利用領先業界的資料管理功能ONTAP、無論資料位於何處、只要使用一組工具、即可管理及保護資料。您也可以自由地將資料移至任何需要的位置：邊緣、核心或雲端。NetApp 9.8.1包含多項功能、可簡化資料管理、加速及保護關鍵資料、並在混合雲架構中提供新一代基礎架構功能。ONTAP

簡化資料管理

資料管理對企業IT營運至關重要、因此應用程式和資料集必須使用適當的資源。包含下列功能、可簡化及簡化作業、並降低總營運成本：ONTAP

- *即時資料壓縮與擴充重複資料刪除技術。*資料壓縮技術可減少儲存區塊內的空間浪費、重複資料刪除技術則可大幅提升有效容量。這適用於本機儲存的資料、以及分層至雲端的資料。
- *服務品質（AQO）的最低、最大及可調適性。*精細的服務品質（QoS）控制、有助於維持高共享環境中關鍵應用程式的效能等級。
- * NetApp FabricPool 功能可將冷資料自動分層至公有雲和私有雲儲存選項、包括Amazon Web Services（AWS）、Azure和NetApp StorageGRID 等儲存解決方案。如需FabricPool 更多有關資訊、請參閱 "[TR-4598](#)"。

加速並保護資料

供應優異的效能與資料保護、並以下列方式擴充這些功能：ONTAP

- *效能與較低的延遲。ONTAP *
- 資料保護 ONTAP 效能*功能提供內建的資料保護功能、並可在所有平台上進行通用管理。
- * NetApp Volume Encryption（NVE）* ONTAP。*支援內建和外部金鑰管理、提供原生Volume層級的加密功能。
- 多租戶與多因素驗證。ONTAP 支援以最高安全層級共享基礎架構資源。

符合未來需求的基礎架構

下列功能可協助滿足嚴苛且不斷變化的業務需求：ONTAP

- 無縫擴充與不中斷營運。ONTAP 支援在不中斷營運的情況下、為現有控制器和橫向擴充叢集增加容量。客戶可以升級至最新技術、例如NVMe和32GB FC、而不需進行昂貴的資料移轉或中斷運作。
- * Cloud connection。ONTAP *效能不只是雲端連線能力最強的儲存管理軟體、ONTAP Select 還可在Cloud Volumes Service 所有公有雲中選擇軟體定義儲存（英文）和雲端原生執行個體（NetApp版本）。
- 與新興應用程式整合 ONTAP。*此功能使用支援現有企業應用程式的相同基礎架構、為新一代平台和應用程式（例如自動駕駛車輛、智慧城市和產業4.0）提供企業級資料服務。

NetApp SANtricity

NetApp SANtricity 旨在為E系列混合式Flash與EF系列All Flash陣列提供領先業界的效能、可靠性與簡易性。針對繁重的工作負載應用、包括資料分析、視訊監控、備份與還原、實現E系列混合式Flash與EF系列All Flash陣列的最大效能與使用率。利用此功能、您可以在儲存設備保持連線的同時、完成組態調整、維護、容量擴充及其他工作。SANtricity此外、透過易於使用的內建System Manager介面、還能提供優異的資料保護、主動監控及認證安全功能。SANtricity若要深入瞭解、請參閱 "[NetApp E系列SANtricity 《軟體資料表》](#)"。

效能最佳化

效能最佳化SANtricity 的功能性支援功能可為您的所有資料分析、視訊監控及備份應用程式、提供高IOPs、高處理量及低延遲的資料。加速高IOPS、低延遲應用程式及高頻寬、高處理量應用程式的效能。

最大化正常運作時間

在儲存設備保持連線的情況下、完成所有管理工作。在不中斷I/O的情況下調整組態、執行維護或擴充容量透過自動化功能、線上組態、最先進的動態磁碟資源池（DPP）技術等功能、實現同級最佳的可靠性。

高枕無憂

透過易於使用的內建System Manager介面、即可提供優異的資料保護、主動監控及認證安全性。SANtricity簡化儲存管理工作。獲得所有E系列儲存系統進階調校所需的靈活度。隨時隨地管理NetApp E系列系統。我們隨裝即用的網路介面可簡化您的管理工作流程。

NetApp Trident

"Trident" NetApp是適用於Docker和Kubernetes的開放原始碼動態儲存協調工具、可簡化持續儲存的建立、管理和使用。Kubernetes原生應用程式Trident直接在Kubernetes叢集內執行。Trident可讓客戶將DL Container映像無縫部署到NetApp儲存設備、並為AI Container部署提供企業級體驗。Kubernetes使用者（例如ML開發人員和資料科學家）可以建立、管理及自動化協調與複製、以充分利用NetApp技術所提供的NetApp進階資料管理功能。

NetApp BlueXP 複製與同步

"BlueXP 複製與同步" 是一項NetApp服務、可快速且安全地同步資料。無論您需要在內部部署 NFS 或 SMB 檔案共用之間傳輸檔案、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 複製與同步功能可快速安全地將檔案移至所需的位置。資料傳輸完成後、即可在來源和目標上完全使用。BlueXP 複製與同步功能會根據預先定義的排程、持續同步資料、只移動資料量、因此資料複製所花的時間和金錢將降至最低。BlueXP 複製與同步是一種軟體即服務（SaaS）工具、設定與使用極為簡單。BlueXP 複製與同步所觸發的資料傳輸是由資料代理人執行。您可以在 AWS、Azure、Google Cloud Platform 或內部部署中部署 BlueXP 複製和同步資料代理人。

Lenovo ThinkSystem伺服器

聯想ThinkSystem伺服器採用創新的硬體、軟體和服務、可解決客戶目前的挑戰、並提供革命性的符合用途的模組化設計方法、以因應未來的挑戰。這些伺服器運用同級最佳的業界標準技術、搭配與眾不同的Lenovo創新技術、為x86伺服器提供最大的靈活度。

部署Lenovo ThinkSystem伺服器的主要優勢包括：

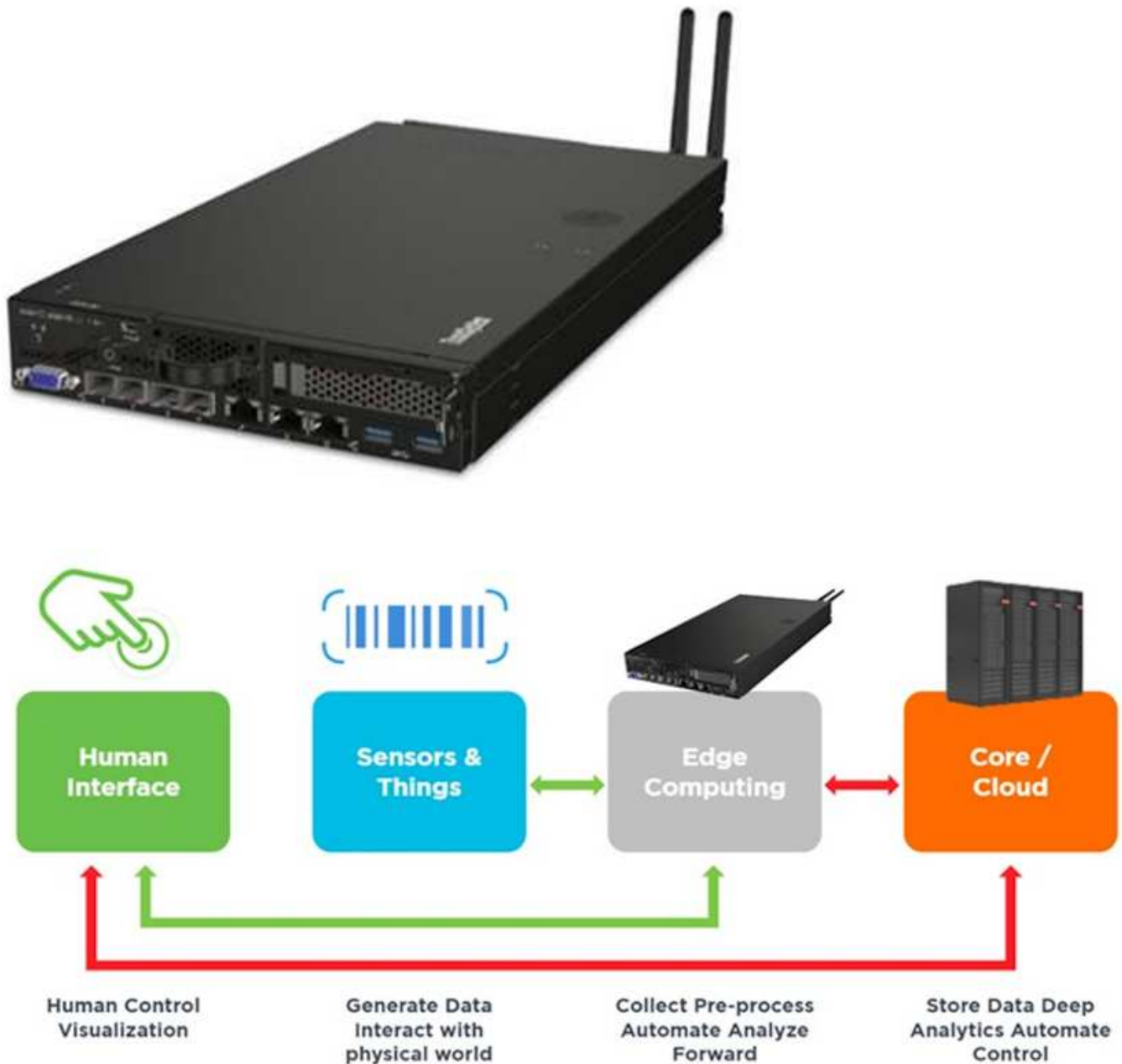
- 可隨企業成長而擴充的模組化設計
- 領先業界的恢復能力、可節省非計畫性停機所需的數小時成本
- 快速Flash技術可降低延遲、縮短回應時間、並即時進行更聰明的資料管理

在AI領域、Lenovo正採取實用的方法、協助企業瞭解ML和AI在工作負載方面的優勢、並將其納入其中。Lenovo客戶可在Lenovo AI創新中心中探索及評估Lenovo AI產品、充分瞭解其特定使用案例的價值。為了縮短實現價值所需的時間、這種以客戶為中心的方法可讓客戶針對已準備好使用並針對AI最佳化的解決方案開發平台、提供概念驗證。

Lenovo ThinkSystem SE350 Edge Server

邊緣運算可讓您在將IoT裝置的資料傳送到資料中心或雲端之前、先在網路邊緣分析資料。下圖所示的Lenovo ThinkSystem SE350、是專為邊緣部署的獨特需求所設計、採用精巧、堅固耐用且環境強化的尺寸規格、著重於靈活度、連線能力、安全性及遠端管理功能。

SE350採用Intel Xeon D處理器、可靈活支援邊緣AI工作負載的加速、專為因應資料中心以外各種環境的伺服器部署挑戰而打造。



MLPerf

MLPerf是領先業界的基準測試套件、可用來評估AI效能。涵蓋多個應用AI領域、包括影像分類、物件偵測、醫療成像和自然語言處理（NLP）。在此驗證中、我們使用推斷v0.7工作負載、這是完成此驗證時最新一次的MLPerf推斷。。 "MLPerf推斷v0.7" 套件包含四項資料中心與邊緣系統的新基準測試：

- 由Transformers（Bert）提供的* Bert.*雙向編碼器表示、可利用Srog資料集微調以回答問題。
- * DLRM-*深度學習建議模式（DLRM）是個人化的推薦模式、經過訓練可最佳化點閱率（CTR）。
- * 3D U-Net.* 3D U-Net架構是針對腦部腫瘤分割（Brat）資料集進行訓練。
- * RNN-T*重複神經網路換能器（RNN-T）是一種自動語音辨識（ASR）模式、訓練內容為LbridSpeech的子

集。MLPerf推斷結果與程式碼均可公開取得、並在Apache授權下發佈。MLPerf推斷具有Edge部門、可支援下列案例：

- *單一串流。*此案例模擬回應能力是關鍵因素的系統、例如在智慧型手機上執行離線AI查詢。個別查詢會傳送至系統、並記錄回應時間。結果會報告所有回應的90個百分位數延遲。
- *多重串流*此基準測試適用於處理多個感應器輸入的系統。在測試期間、查詢會以固定時間間隔傳送。強制實施QoS限制（允許的最大延遲）。測試會報告系統在符合QoS限制的情況下可以處理的串流數量。
- *離線。*這是最簡單的案例、涵蓋批次處理應用程式、而指標是每秒取樣的處理量。系統可取得所有資料、基準測試會測量處理所有樣本所需的時間。

Lenovo已針對本文件所使用的伺服器「T4」（SE350）發表MLPerf推斷分數。請參閱以下網址的結果：
["https://mlperf.org/inference-results-0-7/"](https://mlperf.org/inference-results-0-7/) 在項目#0.7-145的「Edge、Closed Division」一節中。

測試計畫

本文件遵循MLPerf推斷v0.7 "程式碼"（MLPerf推斷v1.1）"程式碼"和 "規則"。我們執行MLPerf基準測試、設計用於在邊緣進行推斷、如下表所定義。

區域	工作	模型	資料集	QSL大小	品質	多重串流延遲限制
願景	映像分類	Resnet50v1.5	ImageNet (224x224)	1024.	99%的FP32	50ms
願景	物件偵測（大型）	SSD - ResNet34	可可（1200 x 1200）	64	99%的FP32	66毫秒
願景	物件偵測（小型）	SSD：移動Netsv1	可可（300x300）	256	99%的FP32	50ms
願景	醫療影像分割	3D UNIT	2019年（全年無休、全年無休、每天24小時160）	16	99%和99.9%的FP32	不適用
演講	語音對文字	RNNT	Lithpech開發-乾淨	2513	99%的FP32	不適用
語言	語言處理	Bert	Squad v1.1	10833	99%的FP32	不適用

下表列出Edge基準測試案例。

區域	工作	案例
願景	映像分類	單一串流、離線、多重串流
願景	物件偵測（大型）	單一串流、離線、多重串流
願景	物件偵測（小型）	單一串流、離線、多重串流
願景	醫療影像分割	單一串流、離線
演講	語音對文字	單一串流、離線
語言	語言處理	單一串流、離線

我們使用此驗證所開發的網路儲存架構來執行這些基準測試、並將結果與先前提交給MLPerf之邊緣伺服器上本

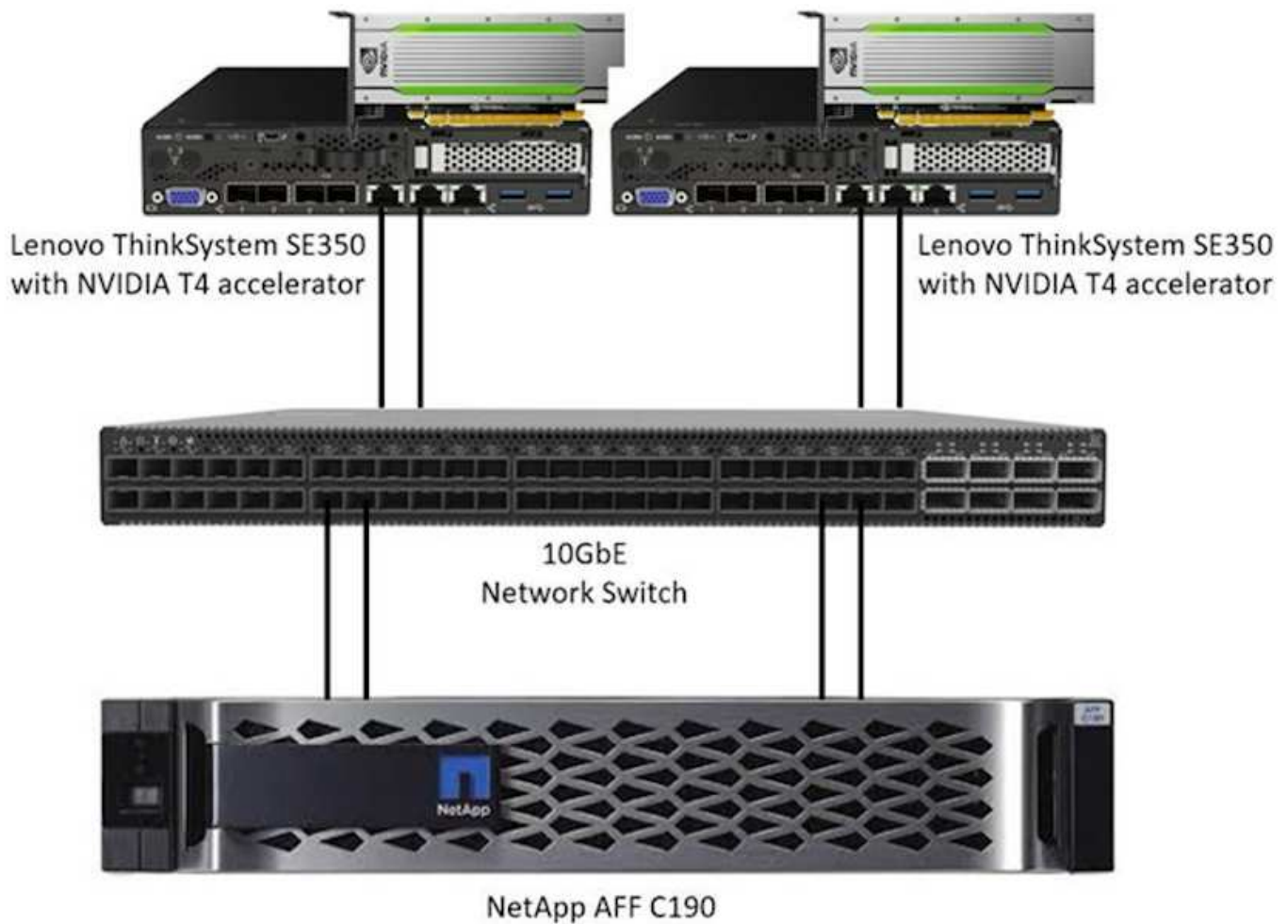
機執行的結果進行比較。比較的目的是判斷共享儲存設備對推斷效能的影響程度。

測試組態

下圖顯示測試組態。我們使用NetApp AFF 的C190儲存系統和兩部Lenovo ThinkSystem SE350伺服器（每部伺服器都有一個NVIDIA T4加速器）。這些元件是透過10GbE網路交換器來連接。網路儲存設備擁有驗證/測試資料集和預先訓練的模型。伺服器提供運算能力、而且儲存設備可透過NFS傳輸協定存取。

本節說明已測試的組態、網路基礎架構、SE350伺服器、以及儲存資源配置詳細資料。下表列出解決方案架構的基礎元件。

解決方案元件	詳細資料
Lenovo ThinkSystem伺服器	<ul style="list-style-type: none">• 2部SE350伺服器各配備1個NVIDIA T4 GPU卡
	<ul style="list-style-type: none">• 每部伺服器都包含一個Intel Xeon D-2123IT CPU、其中四個實體核心執行速度為2.20GHz和128GB RAM
入門級NetApp AFF 產品資料儲存系統（HA配對）	<ul style="list-style-type: none">• NetApp ONTAP ®9軟體• 24個960GB SSD• NFS傳輸協定• 每個控制器一個介面群組、其中四個邏輯IP位址用於掛載點



下表列出儲存組態：AFF 搭載2RU、24個磁碟機插槽的__LW_C190。

控制器	Aggregate	流量FlexGroup	Aggregatesize	體積化	作業系統掛載點
控制器1.	Aggr1.	/netapplenovo_AI_FG	8.42TiB	15TB	/NetApp_聯想_FG
控制器2.	Aggr2.		8.42TiB		

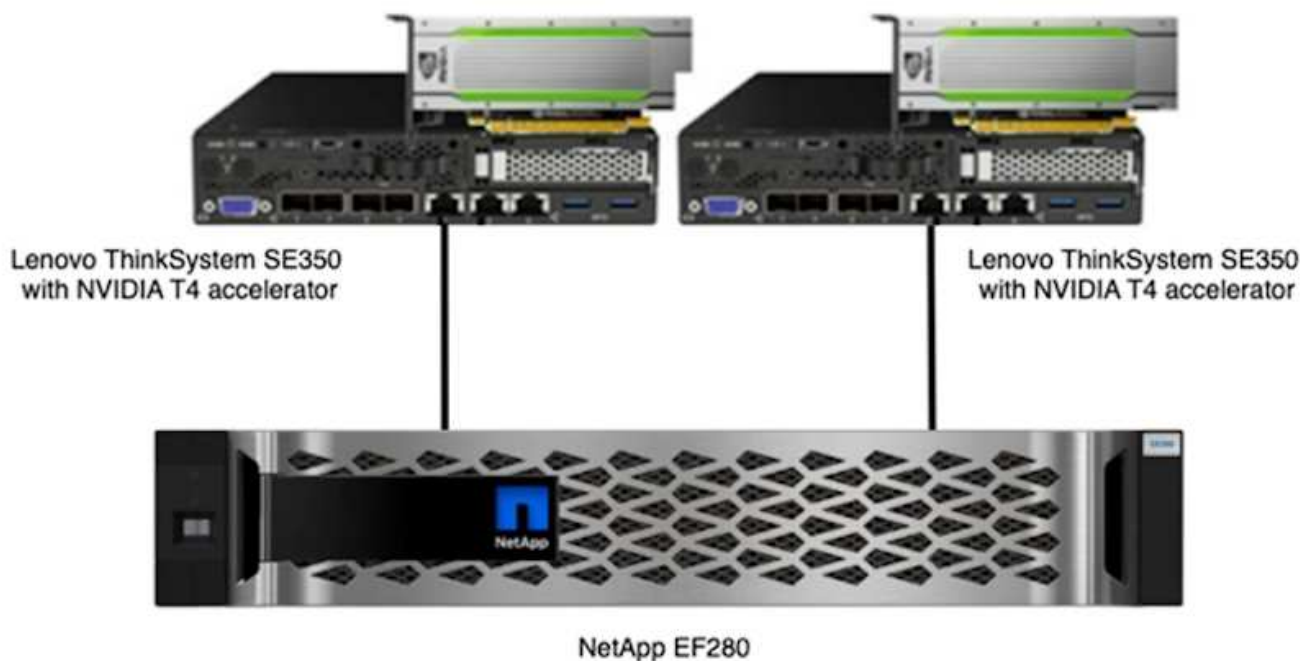
/netapp聯想_AI_fg資料夾包含用於模型驗證的資料集。

下圖顯示測試組態。我們使用NetApp EF280儲存系統和兩個Lenovo ThinkSystem SE350伺服器（每個伺服器都有一個NVIDIA T4加速器）。這些元件是透過10GbE網路交換器來連接。網路儲存設備擁有驗證/測試資料集和預先訓練的模型。伺服器提供運算能力、而且儲存設備可透過NFS傳輸協定存取。

下表列出EF280的儲存組態。

控制器	Volume群組	Volume	體積化	DDPsize	連線方法
控制器1.	DDP1	Volume 1	8.42TiB	16TB	SE350-1至iSCSI LUN 0

控制器	Volume群組	Volume	體積化	DDPsize	連線方法
控制器2.		Volume 2	8.42TiB		SE350-2至iSCSI LUN 1



測試程序

本節說明驗證此解決方案所使用的測試程序。

作業系統與AI推斷設定

在支援NVIDIA GPU的情況下、我們使用Ubuntu 18.04搭配NVIDIA驅動程式和Docker、並使用MLPerf AFF "[程式碼](#)" 在Lenovo提交給MLPerf Inference v0.7的過程中提供。

對於EF280、我們使用Ubuntu 20.04搭配NVIDIA驅動程式和Docker、支援NVIDIA GPU和MLPerf "[程式碼](#)" 在Lenovo提交給MLPerf Inference v1.1的過程中提供。

若要設定AI推斷、請遵循下列步驟：

1. 下載需要登錄的資料集、ImageNet 2012驗證集、Criteo TB資料集、以及Brat 2019訓練集、然後解壓縮檔案。
2. 建立至少1TB的工作目錄、並定義環境變數「MarIFF_Scrate_path」（指目錄）。

您應該在共用儲存設備上共用此目錄、以供網路儲存設備使用、或在測試本機資料時共用本機磁碟。

3. 執行make 「prebuild」 命令、此命令會建立及啟動泊塢視窗容器、以執行必要的推斷工作。



下列命令都是從執行中的Docker容器中執行：

- 下載預先訓練的AI模型以執行MLPerf推斷工作：「make download_model」
- 下載其他可免費下載的資料集：「make download_data」
- 預先處理資料：製作「preprocure_data」
- RUN：「make build」。
- 在運算伺服器中建置專為GPU最佳化的推斷引擎：「make general_引擎」
- 若要執行推斷工作負載、請執行下列（一個命令）：

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

AI推斷會執行

執行了三種類型的執行：

- 使用本機儲存設備的單一伺服器AI推斷
- 使用網路儲存設備的單一伺服器AI推斷
- 使用網路儲存設備的多伺服器AI推斷

測試結果

我們執行多項測試、以評估所建議架構的效能。

共有六種不同的工作負載（影像分類、物件偵測[小]、物件偵測[大]、醫療影像、語音對文字、和自然語言處理[NLP]）、您可以在三種不同的情況下執行：離線、單一串流和多重串流。



最後一個案例只能用於影像分類和物件偵測。

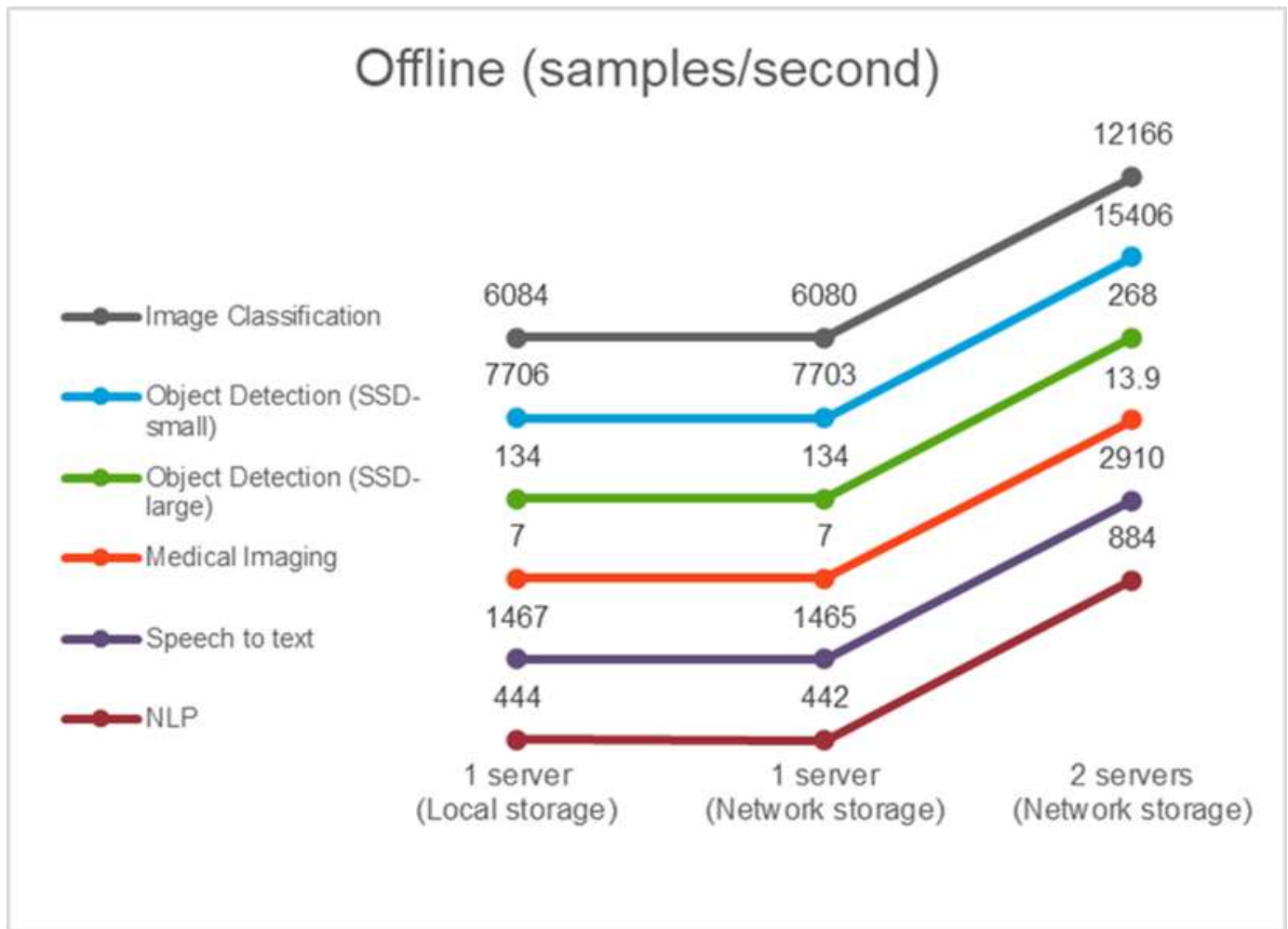
這可提供15種可能的工作負載、這些工作負載都是在三種不同的設定下進行測試：

- 單一伺服器/本機儲存設備
- 單一伺服器/網路儲存設備
- 多伺服器/網路儲存設備

以下各節將說明結果。

AI在離線情境中推斷AFF 以供參考

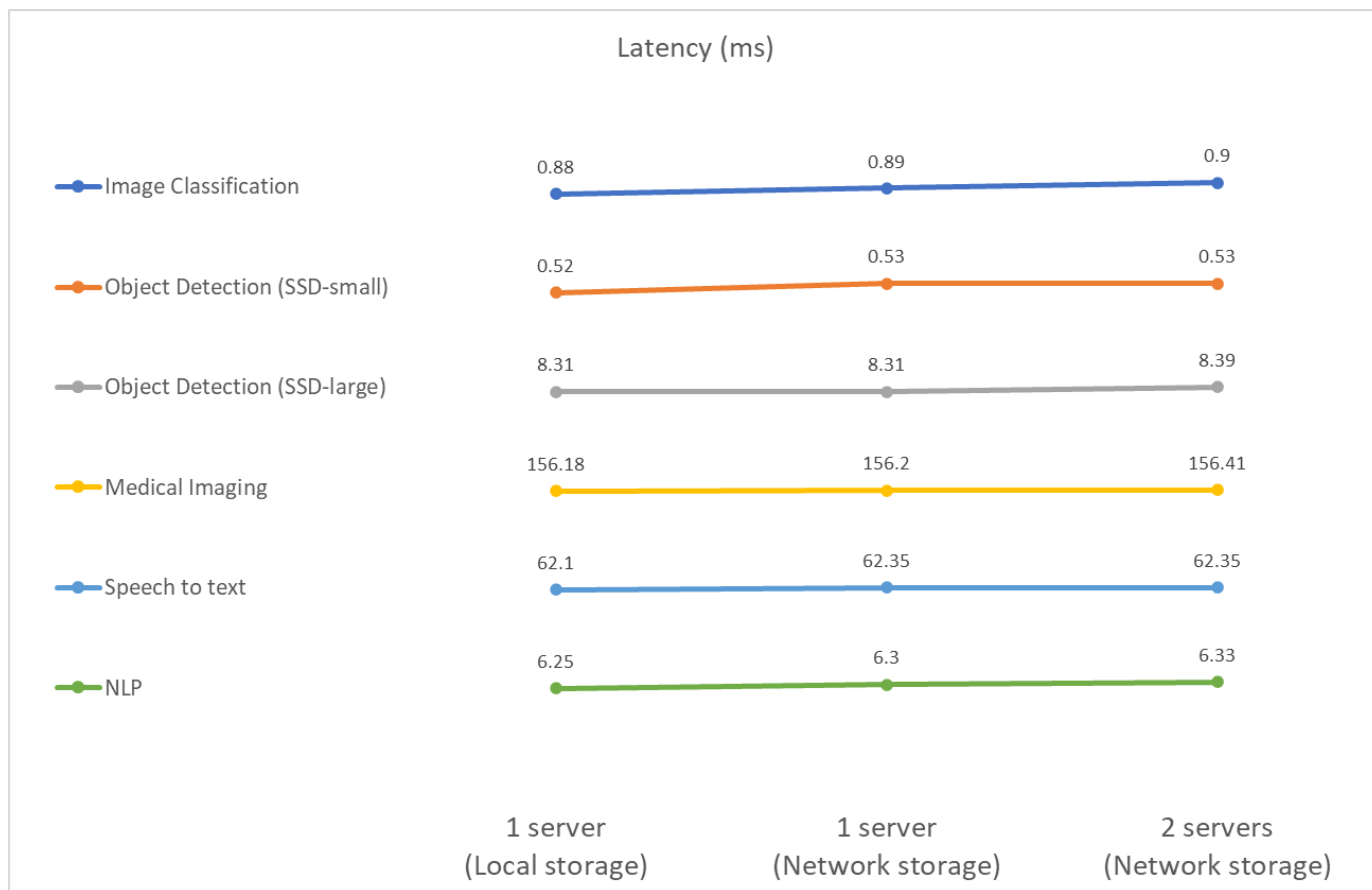
在此案例中、所有資料都可供伺服器使用、並測量處理所有樣本所需的時間。我們會將頻寬報告為每秒取樣數、做為測試結果。當使用多部運算伺服器時、我們會報告所有伺服器的總頻寬。下圖顯示所有三種使用案例的結果。在兩部伺服器的案例中、我們會報告兩部伺服器的總頻寬。



結果顯示、網路儲存設備不會對效能造成負面影響、因為這項變更很小、而且對於某些工作、找不到任何項目。新增第二部伺服器時、總頻寬可能會精確加倍、或是最糟的情況下、變更量會低於1%。

AI在單一串流情境中推斷**AFF** 以供參考

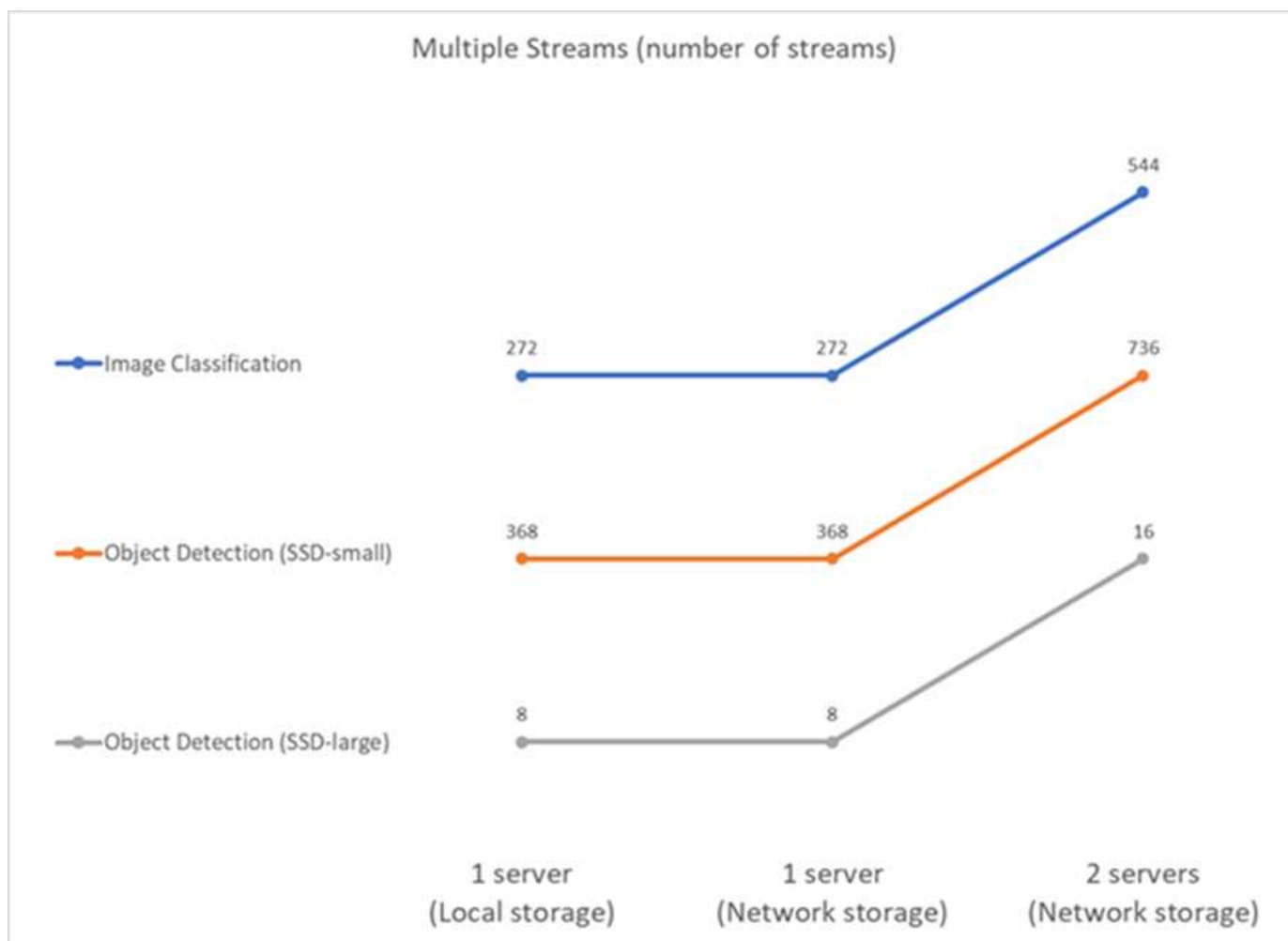
此基準測試可測量延遲。針對多個運算伺服器案例、我們會報告平均延遲。工作套件的結果如下圖所示。在兩部伺服器的案例中、我們會報告兩部伺服器的平均延遲。



結果再次顯示、網路儲存設備足以處理工作。在單一伺服器案例中、本機與網路儲存設備之間的差異最小或無。同樣地、當兩部伺服器使用相同的儲存設備時、兩部伺服器的延遲時間會維持不變、或是變更量會非常小。

AI在多重串流情境中推斷AFF 以供參考

在這種情況下、結果是系統可以處理的串流數量、同時滿足QoS限制。因此、結果永遠是整數。對於多部伺服器、我們會報告在所有伺服器上總計串流的總數。並非所有的工作負載都支援這種情況、但我們已經執行了這些工作負載。我們的測試結果摘要如下圖所示。在兩部伺服器的案例中、我們會報告來自兩部伺服器的串流總數。



結果顯示設定的完美效能：本機與網路儲存設備的結果相同、而新增第二台伺服器的資料流數量是建議設定所能處理的資料流數量的兩倍。

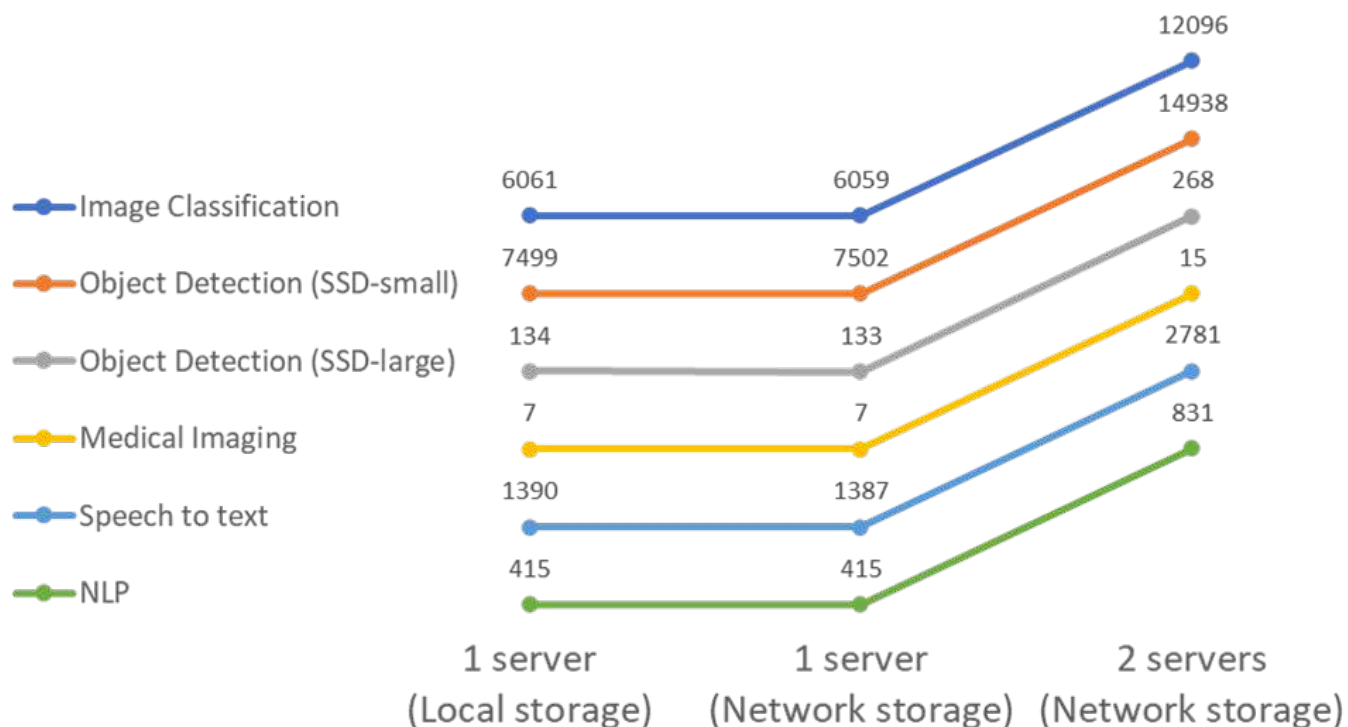
EF的測試結果

我們執行多項測試、以評估所建議架構的效能。共有六種不同的工作負載（影像分類、物件偵測[小]、物件偵測[大]、醫療影像、語音對文字、和自然語言處理（NLP]）、這是在兩種不同的情況下執行：離線和單一串流。以下各節將說明結果。

EF離線情境中的AI推斷

在此案例中、所有資料都可供伺服器使用、並測量處理所有樣本所需的時間。我們會將頻寬報告為每秒取樣數、做為測試結果。對於單一節點執行、我們會報告兩部伺服器的平均執行量、而對於兩部伺服器執行、則會報告所有伺服器的總頻寬總和。使用案例的結果如下圖所示。

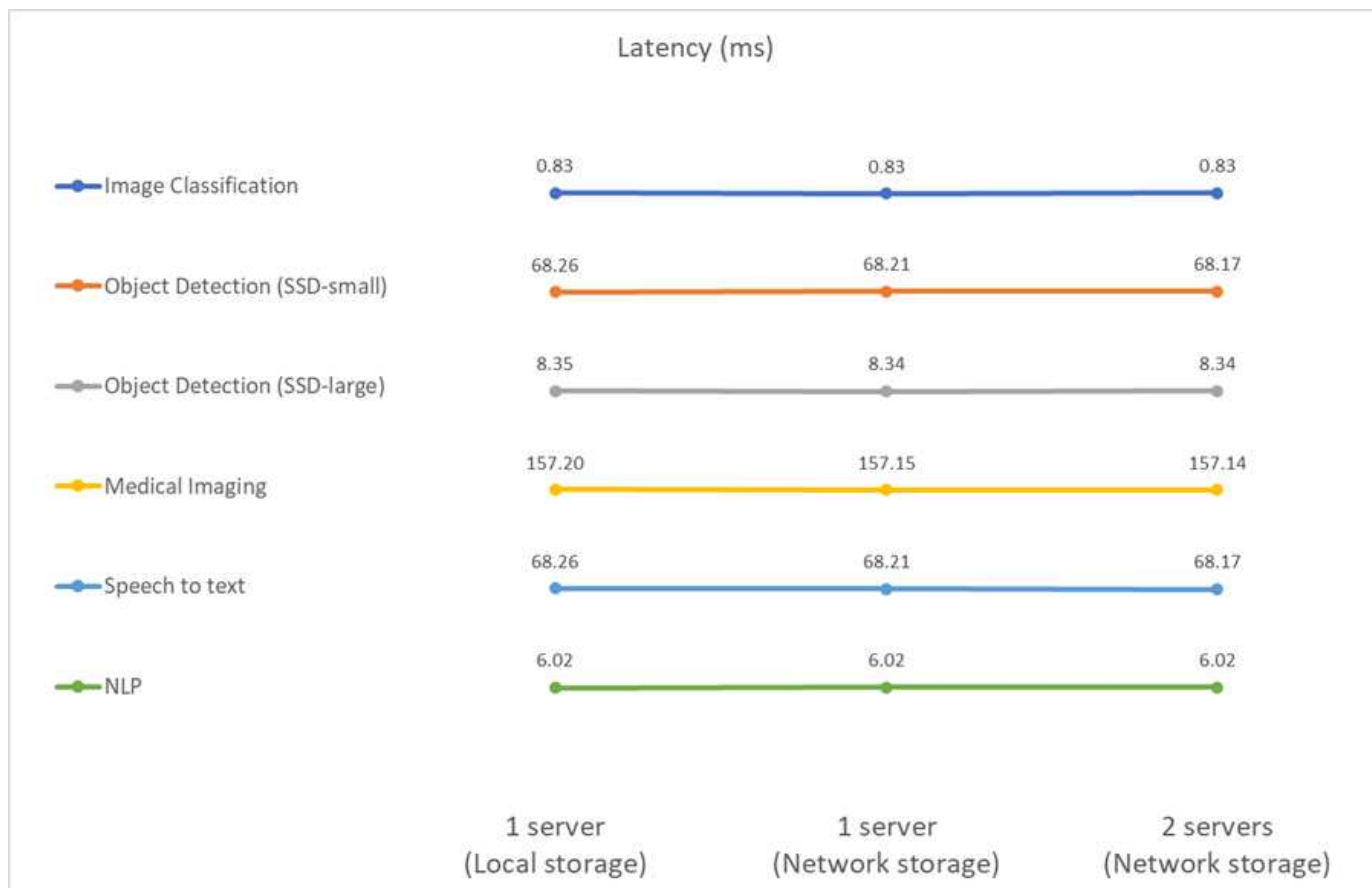
Offline (samples/second)



結果顯示、網路儲存設備不會對效能造成負面影響、因為這項變更很小、而且對於某些工作、找不到任何項目。新增第二部伺服器時、總頻寬可能會精確加倍、或是最糟的情況下、變更量會低於1%。

EF單一串流案例中的AI推斷

此基準測試可測量延遲。在所有情況下、我們都會報告執行中所有伺服器的平均延遲時間。會提供工作套件的結果。



結果再次顯示、網路儲存設備足以處理工作。在單一伺服器的情況下、本機與網路儲存設備之間的差異最小或無。同樣地、當兩部伺服器使用相同的儲存設備時、兩部伺服器的延遲時間會維持不變、或是變量會非常小。

架構規模調整選項

您可以調整用於驗證的設定、以符合其他使用案例。

運算伺服器

我們使用Intel Xeon D-2123IT CPU、這是SE350支援的最低CPU層級、有四個實體核心和60W TDP。雖然伺服器不支援更換CPU、但可以訂購更強大的CPU。支援的最高CPU是Intel Xeon D-2183IT、採用16核心、100W執行速度為2.20GHz。如此可大幅提升CPU的運算能力。雖然CPU本身並不是執行推斷工作負載的瓶頸、但它有助於資料處理及其他與推斷相關的工作。目前、NVIDIA T4是唯一適用於邊緣使用案例的GPU、因此目前無法升級或降級GPU。

共享儲存設備

在測試與驗證方面AFF、NetApp RandomC190系統的儲存容量上限為50.5 TB、循序讀取的處理量為4.4GBps、小型隨機讀取的處理量為230K IOPS、經證實非常適合邊緣推斷工作負載。

不過、如果您需要更多儲存容量或更快的網路速度、您應該使用NetApp AFF 知識庫A220或 ["NetApp AFF 產品-A250"](#) 儲存系統：此外、NetApp EF280系統的最大容量為1.5PB、頻寬為10Gbps、也用於本解決方案驗證。如果您偏好以更高頻寬提供更多儲存容量、["NetApp EF300"](#) 可以使用。

結論

AI導向的自動化與邊緣運算是一項領先業界的方法、可協助企業組織實現數位化轉型、並將作業效率與安全性最大化。利用邊緣運算技術、資料處理速度更快、因為資料中心不需要往返資料中心。因此、將資料來回傳送至資料中心或雲端的相關成本會降低。如果企業必須使用部署在邊緣的AI推斷模型、以近乎即時的方式做出決策、則降低延遲並加快速度將會更為有利。

NetApp儲存系統提供與本機SSD儲存設備相同或更好的效能、並為資料科學家、資料工程師、AI/ML開發人員、企業或IT決策者提供下列效益：

- 輕鬆在AI系統、分析和其他關鍵業務系統之間共享資料。這種資料共享可降低基礎架構的負荷、改善效能、並簡化整個企業的資料管理。
- 獨立擴充的運算與儲存設備、可將成本降至最低、並改善資源使用率。
- 利用整合式Snapshot複本與複製、簡化開發與部署工作流程、提供即時且節省空間的使用者工作區、整合式版本控制、以及自動化部署。
- 企業級的資料保護功能、可實現災難恢復與營運不中斷。本文所述的NetApp與Lenovo解決方案是靈活的橫向擴充架構、非常適合在邊緣部署企業級AI推斷。

感謝

- J.法爾坎格、資深聯想HPC與AI解決方案經理
- NetApp技術行銷工程師Dave Arettette
- NetApp E系列AI解決方案技術主管Joey Parnell
- NetApp品質保證工程師Cody Harman

何處可找到其他資訊

若要深入瞭解本文件所述資訊、請參閱下列文件和/或網站：

- NetApp AFF A系列陣列產品頁面
["https://www.netapp.com/data-storage/aff-a-series/"](https://www.netapp.com/data-storage/aff-a-series/)
- NetApp ONTAP 產品資料管理軟體：ONTAP 9資訊庫
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>
- TR-4727：NetApp EF系列簡介
<https://www.netapp.com/pdf.html?item=/media/17179-tr4727pdf.pdf>
- NetApp E系列SANtricity 產品介紹
<https://www.netapp.com/pdf.html?item=/media/19775-ds-3171-66862.pdf>
- NetApp容器持續儲存設備：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 複製與同步
 - ["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基準測試
 - ["https://github.com/tensorflow/benchmarks"](https://github.com/tensorflow/benchmarks)
- Lenovo ThinkSystem SE350 Edge Server
 - ["https://lenovopress.com/lp1168"](https://lenovopress.com/lp1168)
- Lenovo ThinkSystem DM5100F統一化Flash儲存陣列
 - ["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對話AI

Rick Huang、Sung-HAN LIN、NetApp Davide Onofio、NVIDIA

NVIDIA DGX系統系列是由全球第一款專為企業AI打造的整合式人工智慧（AI）系統所組成。NetApp AFF 供應極致效能與領先業界的混合雲資料管理功能。NetApp與NVIDIA聯手打造NetApp ONTAP 的「可靠性AI參考架構」、這是一套適用於AI與機器學習（ML）工作負載的立即可用解決方案、可提供企業級的效能、可靠性與支援。

本白皮書提供指引、協助客戶建立對話式AI系統、以支援不同產業垂直市場的不同使用案例。其中包含使用NVIDIA JARVIS部署系統的相關資訊。測試是使用NVIDIA DGX Station和NetApp AFF VA220儲存系統執行。

解決方案的目標對象包括下列群組：

- 企業架構設計師、設計解決方案來開發AI模型和軟體、以用於對話式AI使用案例、例如虛擬零售助理
- 尋求有效方法來達成語言建模開發目標的資料科學家
- 負責維護和處理客戶問題和對話記錄等文字資料的資料工程師
- 高階主管與IT決策者及企業領導者、都有興趣從AI計畫轉變對話型AI體驗、並縮短上市時間

解決方案總覽

NetApp ONTAP AI 和 BlueXP 複製與同步

NetApp ONTAP 採用NVIDIA DGX系統和NetApp雲端連線儲存系統的NetApp AI架構、是由NetApp和NVIDIA開發與驗證的。此參考架構可為IT組織提供下列優勢：

- 消除設計複雜性

- 可獨立擴充運算與儲存設備
- 讓客戶從小規模開始、並無縫擴充
- 提供多種儲存選項、適用於各種效能與成本點NetApp ONTAP AI將DGX系統與NetApp AFF 推出先進的網路功能、緊密整合在一起。NetApp ONTAP 採用AI和DGX系統、可免除設計複雜度和猜測、簡化AI部署。客戶可以從小規模開始、以不中斷的方式擴充系統、同時以智慧方式管理從邊緣到核心到雲端及後端的資料。

NetApp BlueXP 複製與同步可讓您輕鬆地在各種通訊協定上移動資料、無論資料是在兩個 NFS 共用區、兩個 CIFS 共用區、或是一個檔案共用區和 Amazon S3 、Amazon Elastic File System (EFS) 或 Azure Blob 儲存區之間。雙主動式作業表示您可以繼續同時使用來源和目標、並在必要時逐步同步資料變更。BlueXP 複製與同步功能可讓您在任何來源與目的地系統之間移動及遞增同步資料、無論是內部部署或雲端型系統、都能以各種新方式使用資料。在內部部署系統、雲端內建與雲端移轉、或是協同作業與資料分析之間移轉資料、都能輕鬆達成。下圖顯示可用的來源和目的地。

在對話式 AI 系統中、開發人員可以利用 BlueXP 複製和同步功能、將雲端的對話記錄歸檔到資料中心、以便離線訓練自然語言處理 (NLP) 模式。透過訓練模式來辨識更多意圖、對話式AI系統將能更妥善地管理終端使用者提出的更複雜問題。

NETAPP JARVIS多模式架構



"NETAPP JARVIS" 是用於建立對話式AI服務的端點對端點架構。其中包括下列GPU最佳化服務：

- 自動語音辨識 (ASR)
- 自然語言理解 (NLU)
- 與網域專屬履約服務整合
- 文字轉換語音 (TTS)
- 以電腦視覺 (CV) JARVIS為基礎的服務採用最先進的深度學習模式、以因應即時對話AI的複雜且具挑戰性的工作。為了與終端使用者進行即時且自然的互動、這些模型需要在300毫秒內完成運算。自然互動是一項挑戰、需要多式模式的感測整合。模型管線也很複雜、需要在上述服務之間進行協調。

JARVIS是一個完全加速的應用程式架構、可用來建置多式對話AI服務、並使用端點對端深度學習管道。JARVIS架構包括預先訓練的對話AI模型、工具、以及針對語音、願景和NLU工作所提供的最佳化端對端服務。除了AI服務之外、JARVIS還能讓您同時融合視覺、音訊及其他感應器輸入、在虛擬助理、多位使用者的資訊化和客服中心助理等應用程式中、提供多重使用者、多重內容對話等功能。

NVIDIA Nemo

"NVIDIA Nemo" 是開放原始碼的Python工具套件、使用易於使用的應用程式設計介面（API）、建置、訓練及微調GPU加速的最新對話AI模型。Nemo使用NVIDIA GPU中的Tensor Cores執行混合式精密運算、可輕鬆擴充至多個GPU、以提供最高的訓練效能。Nemo可用於建置即時ASR、NLP及TTS應用程式的模型、例如視訊通話轉譯、智慧型視訊助理、以及跨不同產業垂直市場（包括醫療、金融、零售及電信）的自動化呼叫中心支援。

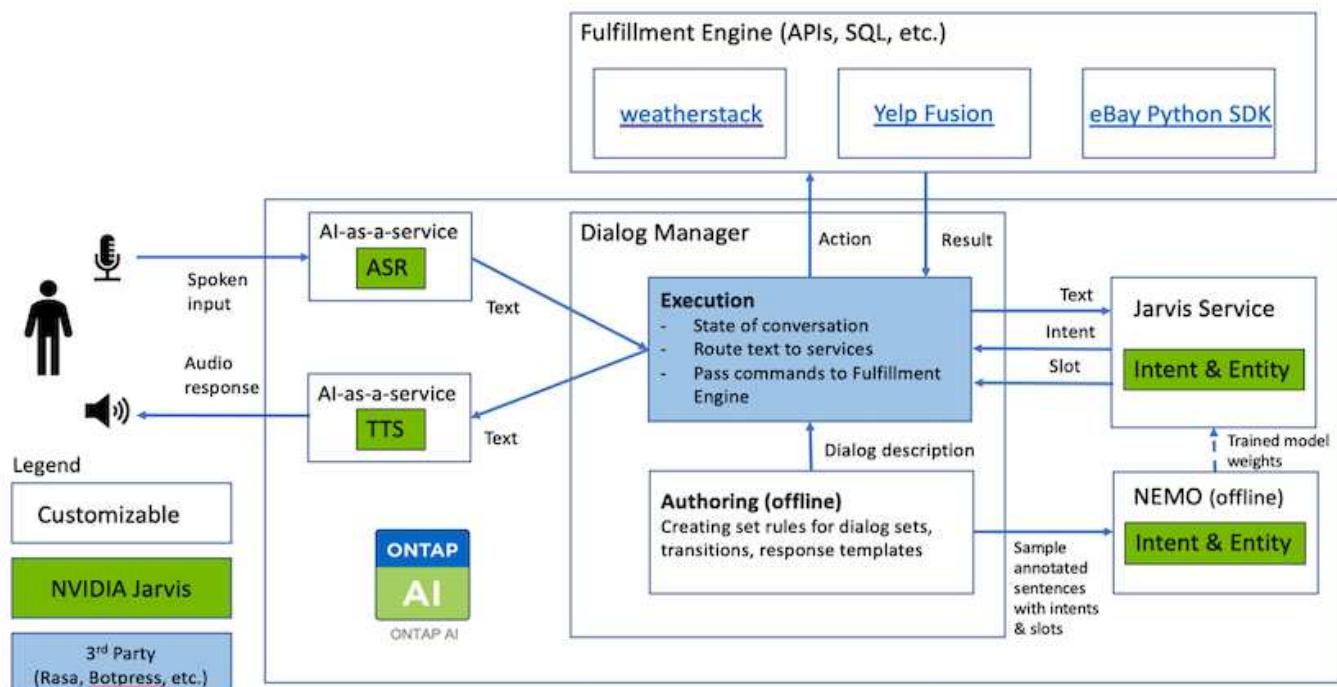
我們使用Nemo來訓練模型、以辨識歸檔對話記錄中使用者問題的複雜意圖。這項訓練將零售虛擬助理的能力延伸到JARVIS所提供的支援之外。

零售使用案例摘要

我們使用NVIDIA JARVIS打造虛擬零售助理、接受語音或文字輸入、並回答有關天氣、興趣點和庫存定價的問題。對話式AI系統能夠記住對話流程、例如、如果使用者未指定天氣或興趣點的位置、請詢問後續問題。系統也能辨識複雜的實體、例如「泰國菜」或「筆記型電腦記憶體」。它瞭解自然語言問題、例如「下週洛杉磯會下雨嗎？」您可以在中找到零售虛擬助理的示範 ["針對零售使用案例自訂狀態和流程"](#)。

解決方案技術

下圖說明提議的對話式AI系統架構。您可以透過語音訊號或文字輸入與系統互動。如果偵測到口頭輸入、則JARVIS AIN-AS服務（AIaaS）會執行ASR、為對話管理程式產生文字。對話管理程式會記住對話狀態、將文字路由傳送至對應的服務、並將命令傳送至Fulfillment Engine。JARVIS NLP服務會輸入文字、辨識意向和實體、並將這些意向和實體插槽輸出回對話方塊管理程式、然後將「行動」傳送至Fulfillment Engine。Fulfillment Engine由第三方API或SQL資料庫組成、可回應使用者查詢。在收到Fulfillment Engine的結果之後、對話管理程式會將文字路由傳送至JARVIS TS IaaS、以產生終端使用者的音訊回應。我們可以將對話記錄歸檔、使用Intents和插槽註釋句子以進行Nemo訓練、讓NLP服務在更多使用者與系統互動時有所改善。



硬體需求

此解決方案已使用一個DGX Station和一個AFF VA220儲存系統進行驗證。JARVIS需要T4或V100 GPU才能執行深度神經網路運算。

下表列出實作已測試解決方案所需的硬體元件。

硬體	數量
T4或V100 GPU	1.
NVIDIA DGX工作站	1.

軟體需求

下表列出實作已測試解決方案所需的軟體元件。

軟體	版本或其他資訊
NetApp ONTAP 資料管理軟體	9.6
Cisco NX-OS交換器韌體	7.0 (3) I6 (1)
NVIDIA DGX OS	4.0.4 - Ubuntu 18.04 LTS
NVIDIA JARVIS架構	企業版0.2
NVIDIA Nemo	nvc.io/nvidia/nemo:v0.10
Docker Container平台	18.06.1-CE [e68fc7a]

總覽

本節詳細說明虛擬零售助理的實作方式。

JARVIS部署

您可以註冊 "[JARVIS早鳥方案](#)" 以存取NVIDIA GPU Cloud (NGC) 上的JARVIS容器。從NVIDIA接收認證之後、您可以使用下列步驟來部署JARVIS：

1. 登入NGC。
2. 將您的組織設在NGC：「ea - 2-jarvis」。
3. 找到JARVIS EA v0.2資產：JARVIS Container位於「Private登錄」>「Organization Container」中。
4. 選取「JARVIS：導覽至「模型指令碼」、然後按一下「JARVIS快速入門」
5. 確認所有資產都能正常運作。
6. 尋找建立您自己應用程式的文件：PDF可在「模型指令碼」>「JARVIS文件」>「檔案瀏覽器」中找到。

針對零售使用案例自訂狀態和流程

您可以針對特定使用案例自訂對話管理程式的狀態和流程。在我們的零售範例中、我們有下列四個yaml檔案、可根據不同的意圖來引導對話。

請參閱下列每個檔案的檔案名稱和說明清單：

- 「MAIN_flow：yaml」：定義主要對話流程和狀態、並在必要時將流程導向其他三個yaml檔案。
- 「REtail_flow。yaml」：包含與零售或興趣點相關的狀態。系統會提供最近的零售店資訊、或是特定項目的價格。
- 「Weather（天氣）」流程：包含與天氣問題相關的狀態。如果無法判斷位置、系統會詢問後續問題以釐清。
- 「error_flow。yaml」：處理使用者意圖不屬於上述三個yaml檔案的案例。顯示錯誤訊息後、系統會重新路由回接受使用者問題。下列各節包含這些yaml檔案的詳細定義。

main_flow

```
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

```

零售_流.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

```

We天氣_流.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

```

錯誤_流.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

```

將第三方API連線為履約引擎

我們將下列第三方API連線為履行引擎、以回答問題：

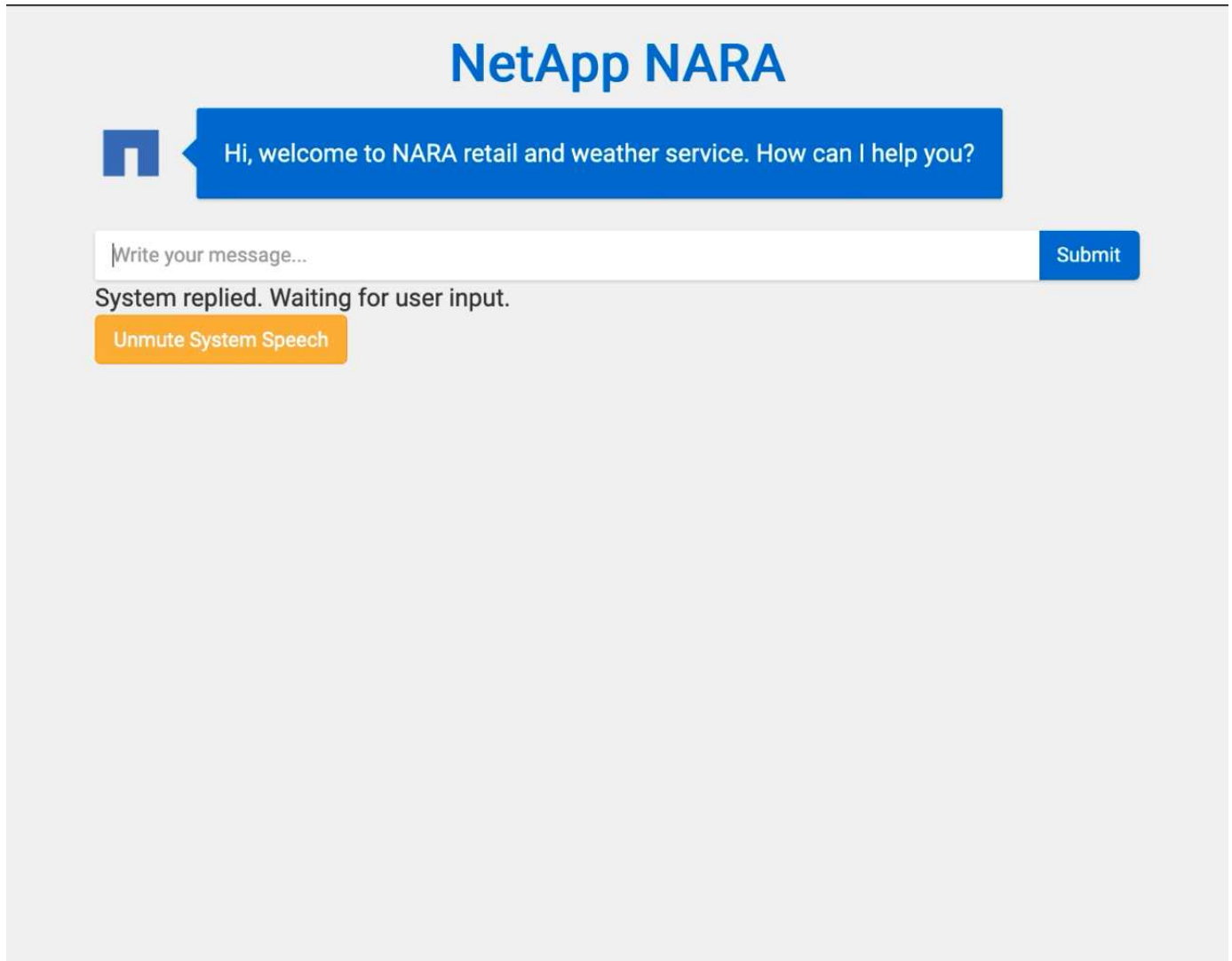
- ["WeatherStack API"](#)：返回給定位置的天氣、溫度、雨和雪。
- ["Yelp Fusion 功能API"](#)：傳回指定位置中最近的商店資訊。
- ["eBay Python SDK"](#)：傳回指定項目的價格。

NetApp 零售助理示範

我們錄製了NetApp零售助理（Nara）的示範影片。

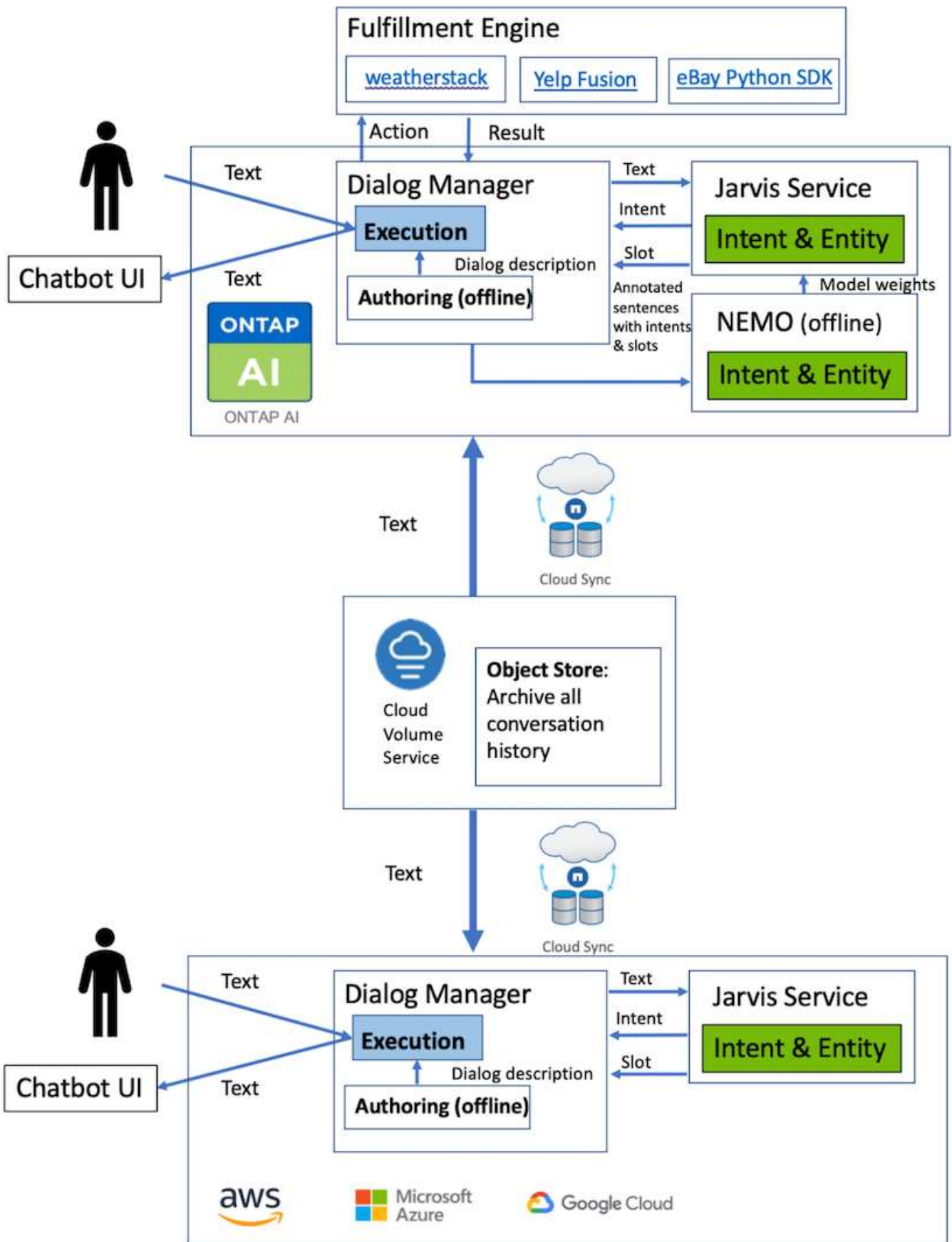
Nara 影片示範

[Nara 影片示範](#)



使用 **NetApp BlueXP** 複製與同步功能來歸檔對話記錄

我們可以每天將對話記錄傾印至 CSV 檔案一次、然後利用 BlueXP 複製與同步功能將記錄檔下載至本機儲存設備。下圖顯示 Jarvis 部署在內部部署和公有雲的架構、同時使用 BlueXP Copy and Sync 傳送 Nemo 訓練的對話記錄。如需Nemo訓練的詳細資料、請參閱一節 "[使用Nemo訓練來擴充Intent模式](#)"。



使用Nemo訓練來擴充Intent模式

NVIDIA Nemo是NVIDIA專為建立對話式AI應用程式所打造的工具套件。此工具套件包含預先訓練的ASR、NLP和TS模組集合、可讓研究人員和資料科學家輕鬆建立複雜的神經網路架構、並將更多焦點放在設計自己的應用程式上。

如上例所示、Nara只能處理有限類型的問題。這是因為預先訓練的NLP模式只會訓練這些類型的問題。如果我們想要讓Nara能夠處理更廣泛的問題、就必須重新訓練自己的資料集。因此、我們在此示範如何使用Nemo來延伸NLP模式、以滿足需求。首先、我們會將從Nara收集的記錄轉換成Nemo格式、然後與資料集一起訓練、以強化NLP模式。

模型

我們的目標是讓Nara根據使用者偏好來排序項目。例如、我們可能會請Nara推薦評價最高的壽司餐廳、或是想要Nara以最低價格尋找這款牛仔褲。為此、我們使用Nemo提供的意向偵測和插槽填滿模式做為訓練模式。此模式可讓Nara瞭解搜尋偏好的意圖。

資料準備

為了訓練模型、我們會收集此類問題的資料集、並將其轉換為Nemo格式。在此列出我們用來訓練模型的檔案。

dict.intents.csv

此檔案列出我們希望Nemo瞭解的所有目標。在此、我們有兩個主要目的和一個意圖、只用於將不符合任何主要目的的問題分類。

```
price_check  
find_the_store  
unknown
```

dict.slots.csv

此檔案列出我們訓練問題上可標示的所有插槽。

```
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
```

訓練：tsv

這是主要的訓練資料集。每一行開頭都是檔案dict.intent.csv中列出的Intent類別之後的問題。從零開始列舉標籤。

訓練插槽.tsv

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

訓練模型

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

然後使用下列命令啟動容器。在此命令中、我們限制容器使用單一GPU（GPU ID = 1）、因為這是輕量化的訓練練習。我們也會將本機工作區/Works/nemo/對應至Container /nemo內的資料夾。

```
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
```

在容器內、如果我們想要從原本訓練好的Bert模型開始、可以使用下列命令來開始訓練程序。data_dir是設定訓練資料路徑的引數。Work目錄可讓您設定儲存檢查點檔案的位置。

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

如果我們有新的訓練資料集、而且想要改善先前的模式、我們可以使用下列命令、從停止點繼續進行。Checkpoint目錄會將路徑移至先前的檢查點資料夾。

```
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
```

推斷模型

我們需要在經過一定次數的時間後、驗證受過訓練的模型的效能。下列命令可讓我們逐一測試查詢。舉例來說、在這個命令中、我們想要檢查我們的模式是否能正確識別查詢「哪裡可以找到最好的義大利麵」的意圖。


```
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
```

接著、以下是推斷的輸出。在輸出中、我們可以看到我們訓練過的模型能夠正確預測future the store的意圖、並傳回我們感興趣的關鍵字。有了這些關鍵字、我們就能讓Nara搜尋使用者想要的內容、並進行更精確的搜尋。

```
[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
```

結論

真正的對話式AI系統會進行類似人的對話、瞭解情境、並提供智慧型回應。這類AI模式通常非常龐大且極為複雜。有了NVIDIA GPU和NetApp儲存設備、就能訓練並最佳化大規模的先進語言模式、以迅速執行推斷。這是邁向結束AI模式的重大進展、而這種模式與大型且複雜的AI模式相比、速度更快。GPU最佳化的語言理解模式可整合至AI應用程式、適用於醫療、零售和金融服務等產業、為智慧型喇叭和客戶服務線中的進階數位語音助理提供支援。這些高品質的對話AI系統可讓垂直市場的企業在與客戶互動時、提供先前無法達成的個人化服務。

JARVIS可部署虛擬助理、數位虛擬助理、多式模式感測Fusion 器等使用案例（CV融合了ASR/NLP/TS）、或任何ASR/NLP/TTS/CV獨立使用案例、例如複本。我們建立了虛擬零售助理、可回答有關天氣、興趣點和庫存定價的問題。我們也示範如何使用 BlueXP 複製與同步歸檔對話記錄、並針對新資料訓練 Nemo 模型、藉此改善對話 AI 系統的自然語言瞭解能力。

感謝

作者非常感謝NVIDIA尊敬的同事們對本白皮書所做的貢獻：Davide Onosfio、Alex Qi、Sicong Ji、Marty Jain和Robert Sohigian。作者也要感謝NetApp主要團隊成員的貢獻

： Santosh Rao 、 David Arnette 、 Michael Oglesby 、 Brent Davis 、 Andy Sayare 、 Erik Minder和Mike McNamara 。

我們衷心感謝所有這些提供見解與專業知識的人員、協助他們建立本白皮書。

何處可找到其他資訊

若要深入瞭解本文所述資訊、請參閱下列資源：

- NVIDIA DGX Station 、 V100 GPU 、 GPU Cloud
 - NVIDIA DGX工作站<https://www.nvidia.com/en-us/data-center/dgx-station/>^[1]
 - NVIDIA V100 Tensor Core GPU<https://www.nvidia.com/en-us/data-center/tesla-v100/>^[2]
 - NVIDIA NGC<https://www.nvidia.com/en-us/gpu-cloud/>^[3]
- NVIDIA JARVIS多模式架構
 - NVIDIA JARVIS<https://developer.nvidia.com/nvidia-jarvis>^[4]
 - NVIDIA JARVIS Early Access<https://developer.nvidia.com/nvidia-jarvis-early-access>^[5]
- NVIDIA Nemo
 - NVIDIA Nemo<https://developer.nvidia.com/nvidia-nemo>^[6]
 - 開發人員指南<https://nvidia.github.io/NeMo/>^[7]
- NetApp AFF 產品
 - NetApp AFF A系列產品型錄<https://www.netapp.com/us/media/ds-3582.pdf>^[8]
 - 適用於All Flash FAS 的NetApp Flash優勢<https://www.netapp.com/us/media/ds-3733.pdf>^[9]
 - 資訊庫ONTAP<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>^[10]
 - NetApp ONTAP FlexGroup 《銷售區》技術報告<https://www.netapp.com/us/media/tr-4557.pdf>^[11]
- NetApp ONTAP AI
 - 《採用DGX-1與Cisco Networking設計指南的AI》ONTAP<https://www.netapp.com/us/media/nva-1121-design.pdf>^[12]
 - 《採用DGX-1和Cisco Networking部署指南的AI》ONTAP<https://www.netapp.com/us/media/nva-1121-deploy.pdf>^[13]
 - 《採用DGX-1和Mellanox網路設計指南的AI》ONTAP<http://www.netapp.com/us/media/nva-1138-design.pdf>^[14]
 - 《採用DGX-2設計指南的AI》ONTAP<https://www.netapp.com/us/media/nva-1135-design.pdf>^[15]

TR-4858：NetApp協調解決方案搭配Run:AI

Rick Huang、David Arignette、Sung-HAN LIN、NetApp Yaron Goldberg、Run：AI

NetApp AFF 供應極致效能與領先業界的混合雲資料管理功能。NetApp與Run：AI合作展示NetApp ONTAP SAI解決方案的獨特功能、適用於人工智慧（AI）與機器學習（ML）工作負載、提供企業級效能、可靠性與支援。執行：AI工作負載的AI協調功能可新增Kubernetes型排程與資源使用平台、協助研究人員管理及最佳化GPU使用率。NetApp、NVIDIA和Run:AI的結合解決方案結合NVIDIA DGX系統、提供專為企業AI工作負載所打造的基礎架構堆疊。本技術報告提供指引、協助客戶建立對話式AI系統、以支援各種使用案例和產業垂直市場。其中包含有關Run：AI和NetApp AFF Sof A800儲存系統部署的資訊、並可做為參考架構、以提供最簡單的方法、協助您快速成功地部署AI方案。

解決方案的目標對象包括下列群組：

- 企業架構設計師設計解決方案、以開發適用於Kubernetes型使用案例的AI模型和軟體、例如容器化微服務
- 資料科學家正在尋求高效率的方法、以便在具有多個團隊和專案的叢集環境中達成有效的模型開發目標
- 負責維護和執行正式作業模式的資料工程師
- 想要創造最佳Kubernetes叢集資源使用體驗、並從AI計畫中獲得最快上市時間的高階主管與IT決策者及企業領導者

解決方案總覽

NetApp ONTAP 的AI與AI控制平面

NetApp ONTAP 與NVIDIA開發並驗證的NetApp AI架構、採用NVIDIA DGX系統與NetApp雲端連線儲存系統。此參考架構可為IT組織提供下列優勢：

- 消除設計複雜性
- 可獨立擴充運算與儲存設備
- 讓客戶從小規模開始、並無縫擴充
- 提供多種儲存選項、可滿足各種效能與成本效益

NetApp ONTAP W人工 智慧將DGX系統與NetApp AFF 的Arde-A800儲存系統與先進的網路技術緊密整合。NetApp ONTAP 採用AI和DGX系統、可免除設計複雜度和猜測、簡化AI部署。客戶可以從小規模開始、以不中斷的方式擴充系統、同時以智慧方式管理從邊緣到核心到雲端及後端的資料。

NetApp AI Control Plane是適用於資料科學家和資料工程師的完整堆疊AI、ML和深度學習（DL）資料與實驗管理解決方案。隨著企業組織增加AI使用率、他們面臨許多挑戰、包括工作負載擴充性和資料可用度。NetApp AI Control Plane透過各種功能來解決這些挑戰、例如像Git repo一樣快速複製資料命名空間、以及定義及實作AI訓練工作流程、將近乎即時的資料建立與模型基準、以利追蹤及版本管理。有了NetApp AI Control Plane、您就能在不同站台和區域之間順暢複寫資料、並迅速配置Jupyter筆記型電腦工作區、存取大量資料集。

Run：AI工作負載協調的AI平台

RUN：AI為AI基礎架構建置了全球第一個協調與虛擬化平台。執行：AI將工作負載從基礎硬體中抽離、建立可動態配置的GPU資源共享資源池、有效協調AI工作負載、並最佳化GPU的使用。資料科學家可以無縫地消耗大

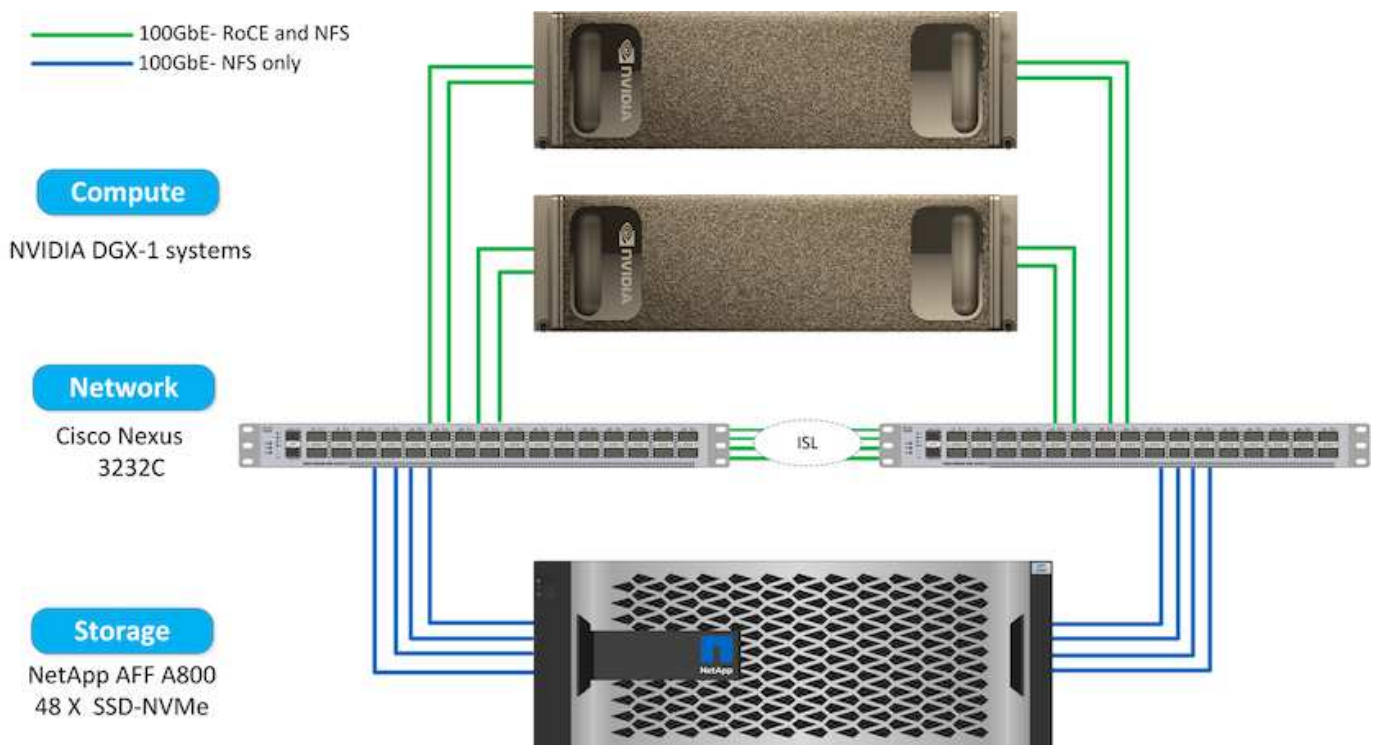
量GPU電力來改善及加速研究、同時IT團隊也能在資源配置、佇列及使用率方面、維持集中化的跨站台控制及即時可見度。Run：AI平台建置於Kubernetes之上、可與現有的IT和資料科學工作流程輕鬆整合。

Run：AI平台提供下列優點：

- *縮短創新時間。*藉由搭配使用Run:AI資源池、佇列及優先順序機制、再搭配NetApp儲存系統、研究人員將不再受基礎架構管理的困擾、而且只能專注於資料科學。RUN：AI與NetApp客戶可在不產生運算或資料傳輸途徑瓶頸的情況下、視需要執行任意數量的工作負載、進而提高生產力。
- *提高團隊生產力。*執行：AI公平演算法可確保所有使用者和團隊都能獲得公平的資源共享。您可以預設優先專案的相關原則、而平台可將資源從一個使用者或團隊動態分配給另一個使用者或團隊、協助使用者及時存取令人夢寐以求的GPU資源。
- 提升**GPU**使用率。Run:AI排程器可讓使用者輕鬆使用部分GPU、整型GPU及GPU的多個節點、進行Kubernetes的分散式訓練。如此一來、AI工作負載就會根據您的需求而非容量來執行。資料科學團隊能夠在相同的基礎架構上執行更多AI實驗。

解決方案技術

此解決方案是以一AFF 部NetApp Efora800系統、兩部DGX-1伺服器、以及兩部Cisco Nexus 3232C 100Gbe-switches來實作。每部DGX-1伺服器均連接至Nexus交換器、並具有四個100GbE連線、可透過融合式乙太網路（RoCE）使用遠端直接記憶體存取（RDMA）進行GPU間通訊。NFS儲存存取的傳統IP通訊也會發生在這些連結上。每個儲存控制器都使用四個100GbE連結連接至網路交換器。下圖顯示ONTAP 本技術報告中針對所有測試案例所使用的AI解決方案架構。



本解決方案所使用的硬體

此解決方案已通過ONTAP 驗證、採用了一套支援整個AI參考架構、其中兩個DGX-1節點和一個AFF 支援完善的資料中心A800儲存系統。請參閱 "[NVA-1121](#)." 如需此驗證所使用基礎架構的詳細資訊、請參閱。

下表列出實作已測試解決方案所需的硬體元件。

硬體	數量
DGX-1系統	2.
解答800 AFF	1.
Nexus 3232C交換器	2.

軟體需求

此解決方案已通過基本Kubernetes部署驗證、並已安裝Run:AI操作員。Kubernetes是使用部署的 ["NVIDIA DeepOps"](#) 部署引擎、可部署正式作業就緒環境所需的所有元件。DeepOps會自動部署 ["NetApp Trident"](#) 為了持續與k8s環境整合儲存設備、我們建立了預設的儲存類別、讓容器能夠利用AFF 來自於該系統的儲存設備。如需ONTAP 有關Trident with Kubernetes on AI的詳細資訊、請參閱 ["TR-4798"](#)。

下表列出實作已測試解決方案所需的軟體元件。

軟體	版本或其他資訊
NetApp ONTAP 資料管理軟體	9.6p4.
Cisco NX-OS交換器韌體	7.0 (3) I6 (1)
NVIDIA DGX OS	4.0.4 - Ubuntu 18.04 LTS
Kubernetes版本	1.17
Trident版本	20.04.0
執行：AI CLI	v2.1.13
RUN：AI Orchestration Kubernetes運算子版本	1.0.39
Docker Container平台	18.06.1-CE [e68fc7a]

有關Run:AI的其他軟體需求、請參閱 ["執行：AI GPU叢集先決條件"](#)。

最佳叢集與GPU使用率搭配Run:AI

下列各節提供執行：AI安裝、測試案例及此驗證所執行結果的詳細資料。

我們使用業界標準基準測試工具（包括TensorFlow基準測試）來驗證此系統的運作和效能。ImageNet資料集用於訓練ResNet-50、這是著名的Convolutional Neuriency Network（CNN/有線新聞網路）DL影像分類模式。ResNet-50提供準確的訓練結果、並加快處理時間、讓我們能夠為儲存設備帶來足夠的需求。

執行：AI安裝

若要安裝Run：AI、請完成下列步驟：

1. 使用DeepOps安裝Kubernetes叢集、並設定NetApp預設儲存類別。
2. 準備GPU節點：
 - a. 確認GPU節點上已安裝NVIDIA驅動程式。

b. 確認「nvidia-Docker」已安裝並設定為預設的泊塢視窗執行時間。

3. 安裝執行：AI：

- 登入 "執行：AI管理UI" 以建立叢集。
- 下載建立的「runai-oper-<clustername>.yaml」檔案。
- 將操作員組態套用至Kubernetes叢集。

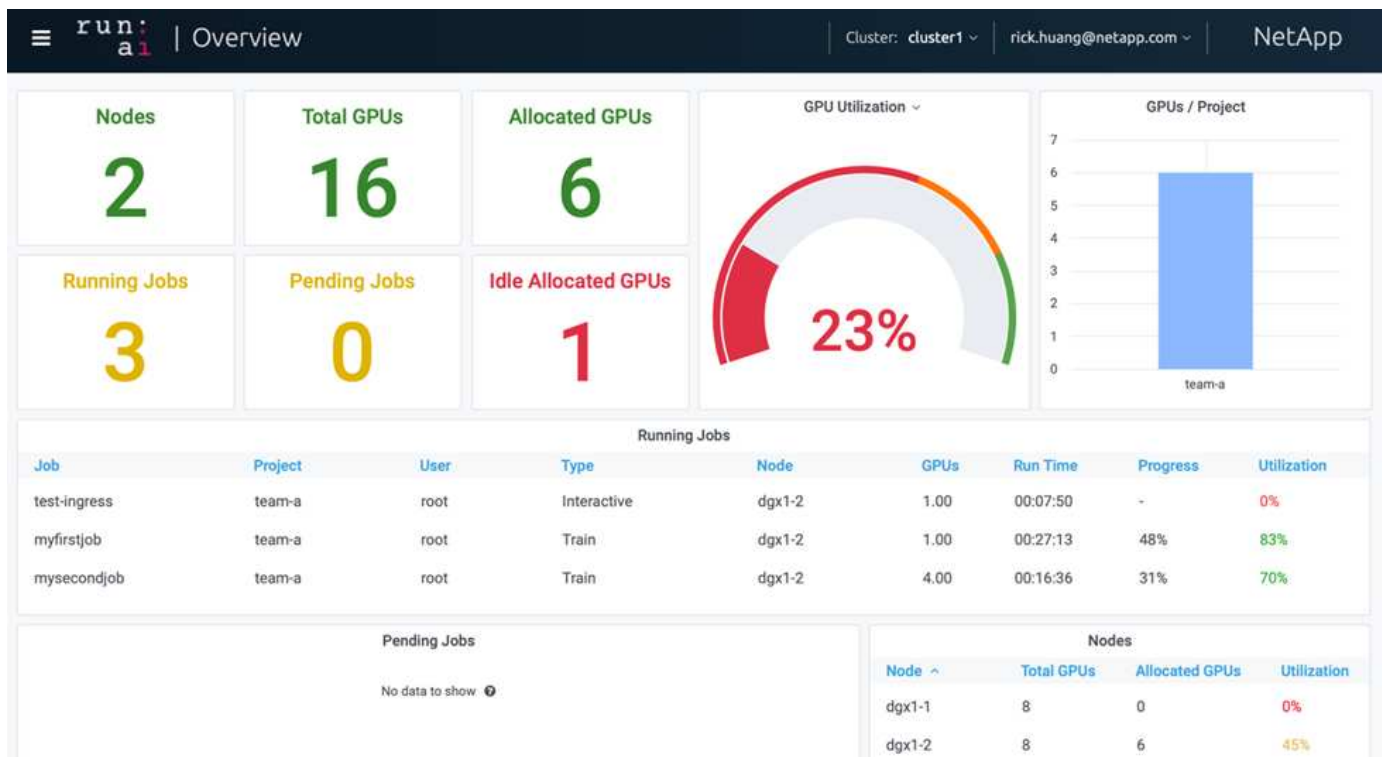
```
kubectl apply -f runai-operator-<clustername>.yaml
```

4. 驗證安裝：

- 前往 "<https://app.run.ai/>"。
- 前往「總覽」儀表板。
- 確認右上角的GPU數量反映出GPU的預期數量、而GPU節點全都列在伺服器清單中。如需執行：AI部署的詳細資訊、請參閱 "[在內部部署Kubernetes叢集上安裝Run:AI](#)" 和 "[安裝Run:AI CLI](#)"。

執行：AI儀表板和檢視

在Kubernetes叢集上安裝Run:AI並正確設定容器之後、您會在上看到下列儀表板和檢視 "<https://app.run.ai/>" 下圖所示。



叢集中共有16個GPU、由兩個DGX-1節點提供。您可以查看節點數量、可用的GPU總數、指派給工作負載的已分配GPU、執行中工作的總數、擱置中工作、以及閒置配置的GPU。右側的長條圖顯示每個專案的GPU、其中摘要說明不同的團隊如何使用叢集資源。中間是目前執行中工作的清單、其中包含工作詳細資料、包括工作名稱、專案、使用者、工作類型、每個工作所在的節點、為該工作分配的GPU數量、工作目前的執行時間、工作進度百分比、以及該工作的GPU使用率。請注意、叢集使用率偏低（GPU使用率為23%）、因為單一團隊

(「team A」) 只提交三個執行中工作。

在下一節中、我們將示範如何在「專案」索引標籤中建立多個團隊、並為每個團隊分配GPU、以便在每個叢集有許多使用者時、將叢集使用率最大化並管理資源。測試案例模擬企業環境、在這些環境中、訓練、推斷及互動式工作負載之間共享記憶體與GPU資源。

為Data科學團隊建立專案並分配GPU

研究人員可以透過Run:AI CLI、Kubeflow或類似程序來提交工作負載。為了簡化資源配置並建立優先順序、Run:AI引進專案概念。專案是配額實體、可將專案名稱與GPU配置和偏好設定建立關聯。這是管理多個資料科學團隊的簡易方法。

提交工作負載的研究人員必須將專案與工作負載要求建立關聯。Run:AI排程器會將要求與目前的配置和專案進行比較、並決定是否可以分配資源給工作負載、或是否應保持擱置狀態。

身為系統管理員、您可以在「執行：AI專案」索引標籤中設定下列參數：

- ***模型專案。***設定每個使用者的專案、設定每個使用者小組的專案、以及針對真正的組織專案設定專案。
- ***專案配額。***每個專案都會與GPU配額相關聯、以便同時分配給此專案。這是一個保證配額、因為無論叢集的狀態為何、使用此專案的研究人員都能獲得此數量的GPU。一般而言、專案配置的總和應等於叢集中的GPU數量。除此之外、此專案的使用者可能會收到過度配額。只要未使用GPU、使用此專案的研究人員就能取得更多GPU。我們在中示範過配額測試案例和公平考量 ["透過過度配額GPU配置、實現高叢集使用率"](#)、["基本資源配置公平性"](#)和 ["過度配額的公平性"](#)。
- 建立新專案、更新現有專案、以及刪除現有專案。
- 限制在特定節點群組上執行工作。您可以指派特定專案、僅在特定節點上執行。當專案團隊需要專用的硬體（例如有足夠的記憶體）時、此功能非常實用。或者、專案團隊可能是以專業預算購買的特定硬體擁有者、或是當您需要將建置或互動式工作負載導向較弱的硬體、並將較長的訓練或無人管理的工作負載導向較快的節點時。如需群組節點及設定特定專案關聯性的命令、請參閱 ["執行：AI文件"](#)。
- 限制互動工作的持續時間。研究人員經常忘了關閉互動工作。這可能會導致資源浪費。有些組織偏好限制互動工作的持續時間、並自動關閉。

下圖顯示建立了四個團隊的專案檢視。每個團隊都會指派不同數量的GPU來處理不同的工作負載、GPU總數等於由兩個DGX-1組成的叢集中可用GPU總數。

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
<div><div></div><div>team-a</div></div>	2	07/27/20, 9:28AM	none	none
<div><div></div><div>team-b</div></div>	4	07/28/20, 7:50AM	none	none
<div><div></div><div>team-c</div></div>	2	07/28/20, 7:50AM	none	none
<div><div></div><div>team-d</div></div>	8	07/28/20, 7:51AM	none	none

在Run:AI CLI中提交工作

本節提供基本Run:AI命令的詳細資訊、可用於執行任何Kubernetes工作。它根據工作負載類型分為三個部分。AI / ML / DL工作負載可分為兩種一般類型：

- 無人蔘與的訓練課程。有了這些類型的工作負載、資料科學家就能準備好自行執行工作負載、然後將其傳送執行。執行期間、客戶可以檢查結果。這類工作負載通常用於正式作業、或是在不需要人為介入的階段進行模型開發。
- 互動式建置工作階段。有了這些類型的工作負載、資料科學家就能開啟與Bash、Jupyter Notebook、遠端PyCharm或類似的IDE互動式工作階段、並直接存取GPU資源。我們在第三個案例中、使用連接埠執行互動式工作負載、以向容器使用者展示內部連接埠。

無人管理的訓練工作負載

設定專案並分配GPU之後、您可以在命令列使用下列命令來執行任何Kubernetes工作負載：

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

此命令會為團隊A開始無人值守的訓練工作、並分配單一GPU。工作是以「GCR.IO/RUN -AI DEMO / quickstart」樣本泊塢視窗影像為基礎。我們將工作命名為「hyper1」。然後您可以執行下列命令來監控工作的進度：

```
$ runai list
```

下圖顯示「Runai list」命令的結果。您可能會看到下列一般狀態：

- 「ContainerCreating」。Docker容器正在從雲端儲存庫下載。
- 「待處理」。工作正在等待排程。
- 《執行中》。工作正在執行中。

```
~> 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
```

若要取得工作的其他狀態、請執行下列命令：

```
$ runai get hyper1
```

若要檢視工作記錄、請執行「Runai logs <job-name>」命令：

```
$ runai logs hyper1
```

在此範例中、您應該會看到正在執行的DL工作階段記錄、包括目前的訓練時期、ETA、Loss Function Value

、Accuracy、以及每個步驟所經過的時間。

您可以在Run:AI UI上檢視叢集狀態 "<https://app.run.ai/>"。在「儀表板」>「總覽」下、您可以監控GPU使用率。

若要停止此工作負載、請執行下列命令：

```
$ runai delete hyper1
```

此命令可停止訓練工作負載。您可以再次執行「Runai list」（Runai清單）來驗證此動作。如需詳細資訊、請參閱 "[啟動無人管理的訓練工作負載](#)"。

互動式建置工作負載

設定專案並分配GPU之後、您可以在命令列使用下列命令來執行互動式建置工作負載：

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

工作是以範例Docker影像python為基礎。我們將工作建置命名為「1」。



「互動」旗標表示工作沒有開始或結束研究人員有責任完成這項工作。系統管理員可以定義互動工作的時間限制、之後系統會終止這些工作。

「-g 1」旗標會將單一GPU配置給此工作。提供的命令與引數是「-command sleep- args infinity」。您必須提供命令、否則容器會立即啟動並結束。

下列命令的運作方式與中所述的命令類似 [\[無人管理的訓練工作負載\]](#)：

- 「Runai list」：顯示名稱、狀態、年齡、節點、映像、專案、使用者及GPU的工作。
- 「Runai Get build1」：在工作建置1上顯示其他狀態。
- "Runai DELETE build1"：停止互動式工作負載建置1。若要將Bash Shell移至容器、請執行下列命令：

```
$ runai bash build1
```

這可直接在電腦中提供Shell。然後、資料科學家可以在容器內開發或微調模型。

您可以在Run:AI UI上檢視叢集狀態 "<https://app.run.ai/>"。如需詳細資訊、請參閱 "[啟動及使用互動式建置工作負載](#)"。

互動式工作負載與連接的連接埠

作為互動式建置工作負載的延伸、您可以在使用Run:AI CLI啟動容器時、向容器使用者顯示內部連接埠。這對於雲端環境、使用Jupyter筆記型電腦或連線至其他微服務都很有用。"入侵" 允許從Kubernetes叢集外部存取Kubernetes服務。您可以建立規則集合來定義哪些傳入連線可到達哪些服務、藉此設定存取。

為了更妥善管理叢集中的外部服務存取、建議叢集管理員安裝 "入侵" 並設定負載平衡器。

若要使用Ingress做為服務類型、請執行下列命令、在提交工作負載時設定方法類型和連接埠：

```
$ 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
```

容器成功啟動後、執行「Runai list」（執行清單）以查看用來存取Jupyter Notebook的「服務URL（S）」。URL由入口端點、工作名稱和連接埠組成。例如、請參閱 <https://10.255.174.13/test-ingress-8888>。

如需詳細資料、請參閱 "使用連接的連接埠啟動互動式建置工作負載"。

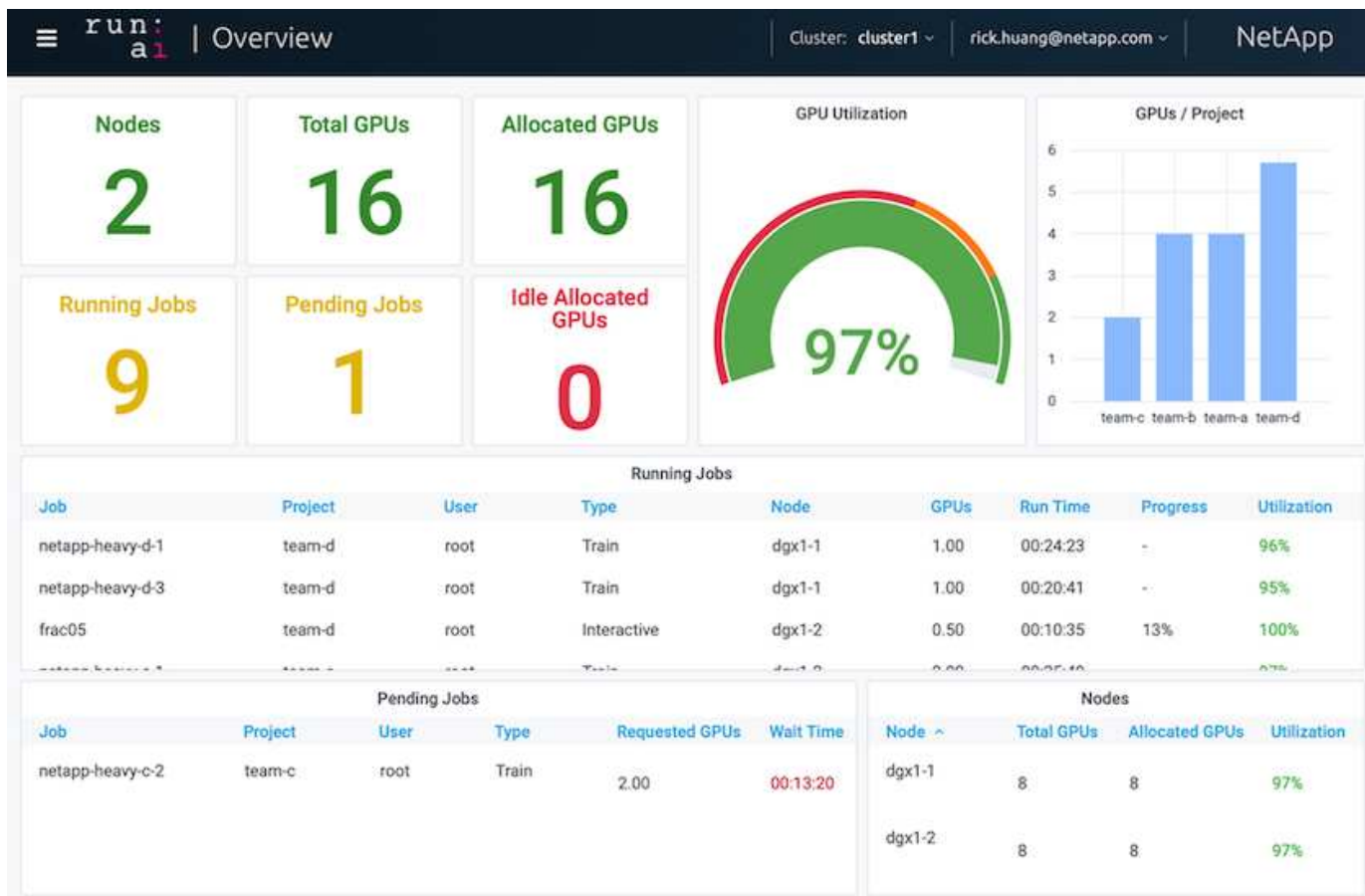
實現高叢集使用率

在本節中、我們模擬一個實際案例、其中四個資料科學團隊各自提交自己的工作負載、以展示Run:AI協調解決方案、在維持優先順序和平衡GPU資源的同時、達到高叢集使用率。我們首先使用一節中所述的ResNet-50基準測試 "ResNet-50搭配ImageNet資料集基準測試摘要"：

```
$ 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
```

我們執行的ResNet-50基準測試與中相同 "NVA-1121"。我們使用旗標「-本機映像」來表示未駐留在公用泊塢視窗儲存庫中的容器。我們將主機DGX-1節點上的目錄「/mnt」和「/tmp」分別掛載到容器上。資料集位於NetApp AFFA800、其中「dataset_dir」引數指向目錄。無論是「-num_devices=1」或「-g 1」、我們都會為此工作分配一個GPU。前者是所謂的「run.py」指令碼、後者則是「runai submit」指令的旗標。

下圖顯示系統總覽儀表板、其中GPU使用率達97%、並已配置全部16個可用GPU。您可以在GPU /專案長條圖中輕鬆查看每個團隊分配的GPU數量。「執行中工作」窗格會顯示目前執行中的工作名稱、專案、使用者、類型、節點、GPU使用量、執行時間、進度和使用率詳細資料。佇列中的工作負載與其等待時間清單、會顯示在「擱置工作」中。最後、節點方塊會針對叢集中的個別DGX-1節點、提供GPU數量和使用率。



部分GPU配置、適用於要求較低或互動性較差的工作負載

當研究人員和開發人員在開發、超參數調校或偵錯階段、都在研究他們的模型時、這類工作負載通常需要較少的運算資源。因此、配置部分GPU和記憶體的效率較高、因此同一個GPU可同時分配給其他工作負載。RUN：AI的協調化解決方案為Kubernetes上的容器化工作負載提供部分GPU共享系統。系統支援執行CUDA程式的工作負載、特別適合輕量化AI工作、例如推斷和建構模型。部分GPU系統可讓資料科學和AI工程團隊在單一GPU上同時執行多個工作負載。如此一來、公司就能在同一個硬體上執行更多工作負載、例如電腦視覺、語音辨識和自然語言處理、進而降低成本。

RUN：AI的部分GPU系統可有效建立虛擬化邏輯GPU、並提供其專屬的記憶體與運算空間、讓容器如同獨立的處理器一樣使用及存取。如此一來、多個工作負載就能在同一個GPU的容器中並排執行、而不會互相干擾。此解決方案透明、簡單且可攜、不需變更容器本身。

典型的usecase可以看到在同一個GPU上執行兩到八個工作、也就是說、您可以使用相同的硬體來執行八倍的工作。

從下圖中的項目"team d (團隊d)"的"Fract05 (分裂05)"工作中可以看出，分配給GPU的數量是0.5。「nvidia-smi」命令可進一步驗證、顯示容器可用的GPU記憶體為16255MB：DGX-1節點每V100 GPU 32GB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 |
+-----+

```

透過過度配額GPU配置、實現高叢集使用率

在本節和各節中 "[基本資源配置公平性](#)"和 "[過度配額的公平性](#)"我們設計了進階測試案例、以展示Run:AI協調功能、適用於複雜的工作負載管理、自動優先排程、以及過度配額GPU資源配置。我們這樣做是為了在ONTAP 整個AI環境中、達到高叢集資源使用率、並最佳化企業級資料科學團隊的生產力。

在這三個區段中、設定下列專案和配額：

專案	配額
團隊A	4.
團隊b	2.
團隊	2.
團隊	8.

此外、這三個區段還使用下列容器：

- Jupyter Notebook：《jupyter/base筆記型電腦》
- RUN：AI quickstart：「GCR.IO/RUN-AI-DEMO / quickstart」

我們為此測試案例設定下列目標：

- 展現資源配置的簡易性、以及如何從使用者中抽取資源

- 示範使用者如何輕鬆配置GPU的一部分和GPU的整數數目
- 示範系統如何排除運算瓶頸、讓團隊或使用者在叢集中有可用的GPU時、可以跳過資源配額
- 示範如何在執行運算密集工作（例如NetApp容器）時、使用NetApp解決方案來消除資料管線瓶頸
- 示範多種容器如何使用系統執行
 - Jupyter筆記型電腦
 - 執行：AI Container
- 叢集已滿時顯示高使用率

如需測試期間實際執行命令順序的詳細資料、請參閱 ["第4.8節的測試詳細資料"](#)。

提交所有13項工作負載時、您會看到已分配的容器名稱和GPU清單、如下圖所示。我們有七項訓練和六項互動工作、模擬四個資料科學團隊、每個團隊都有自己的執行或開發模式。對於互動工作、個別開發人員使用Jupyter Notebooks來撰寫或偵錯其程式碼。因此、它適合在不使用太多叢集資源的情況下配置GPU分數。

```
root@run-deploy:~# kubectl get pods -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

此測試案例的結果顯示下列項目：

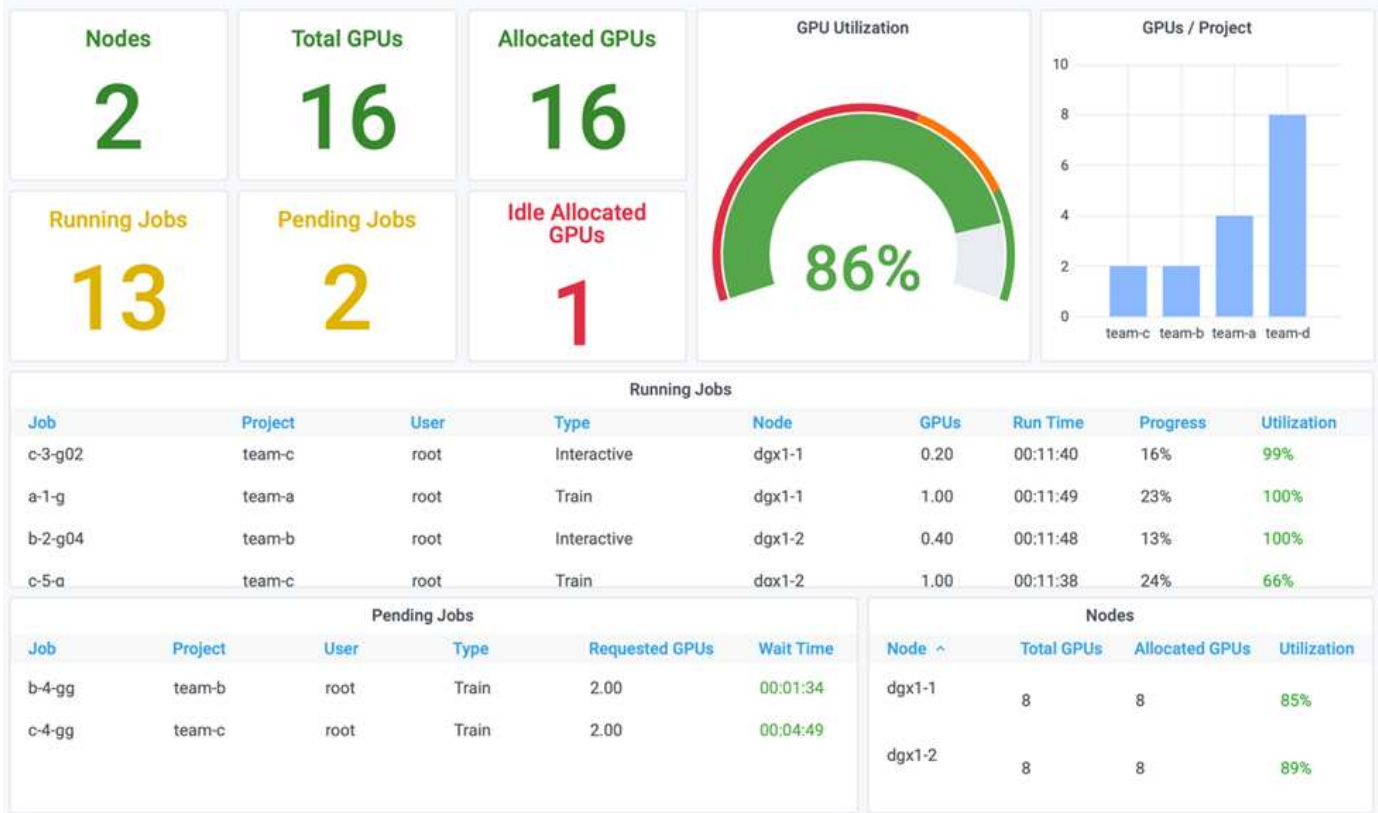
- 叢集應已滿：使用16/16 GPU。
- 叢集使用率高。
- 由於分數分配、實驗量比GPU多。
- 因為，“Team d”並沒有全部使用配額，所以“team-b”和“team-c”可以使用額外的GPU進行實驗，從而縮短創新時間。

基本資源配置公平性

在本節中、我們顯示、當「team d」要求更多GPU（以配額為限）時、系統會暫停「team b」和「team c」的工作負載、並以公平的共享方式將其移至待處理狀態。

如需工作提交、使用的容器映像及執行命令順序等詳細資料、請參閱一節 ["第4.9節的測試詳細資料"](#)。

下圖顯示所產生的叢集使用率、每個群組分配的GPU、以及由於自動負載平衡和優先排程而產生的擱置工作。我們可以觀察到、當所有團隊工作負載所要求的GPU總數超過叢集中可用的GPU總數時、Run:AI的內部公平演算法會因為達到專案配額、而在「team b」和「team -c」中各暫停一項工作。這可提供整體高叢集使用率、而資料科學團隊仍在系統管理員設定的資源限制下工作。



此測試案例的結果顯示下列項目：

- *自動負載平衡。*系統會自動平衡GPU的配額、使每個團隊現在都使用配額。暫停的工作負載屬於超出配額的團隊。
- *公平共用暫停。*系統會選擇停止某個團隊的工作負載、使其超出配額、然後停止另一個團隊的工作負載。RUN：AI具有內部公平演算法。

過度配額的公平性

在本節中、我們將展開多個團隊提交工作負載並超過其配額的案例。如此一來、我們就能示範Run：AI的公平演算法如何根據預設配額的比率來配置叢集資源。

本測試案例的目標：

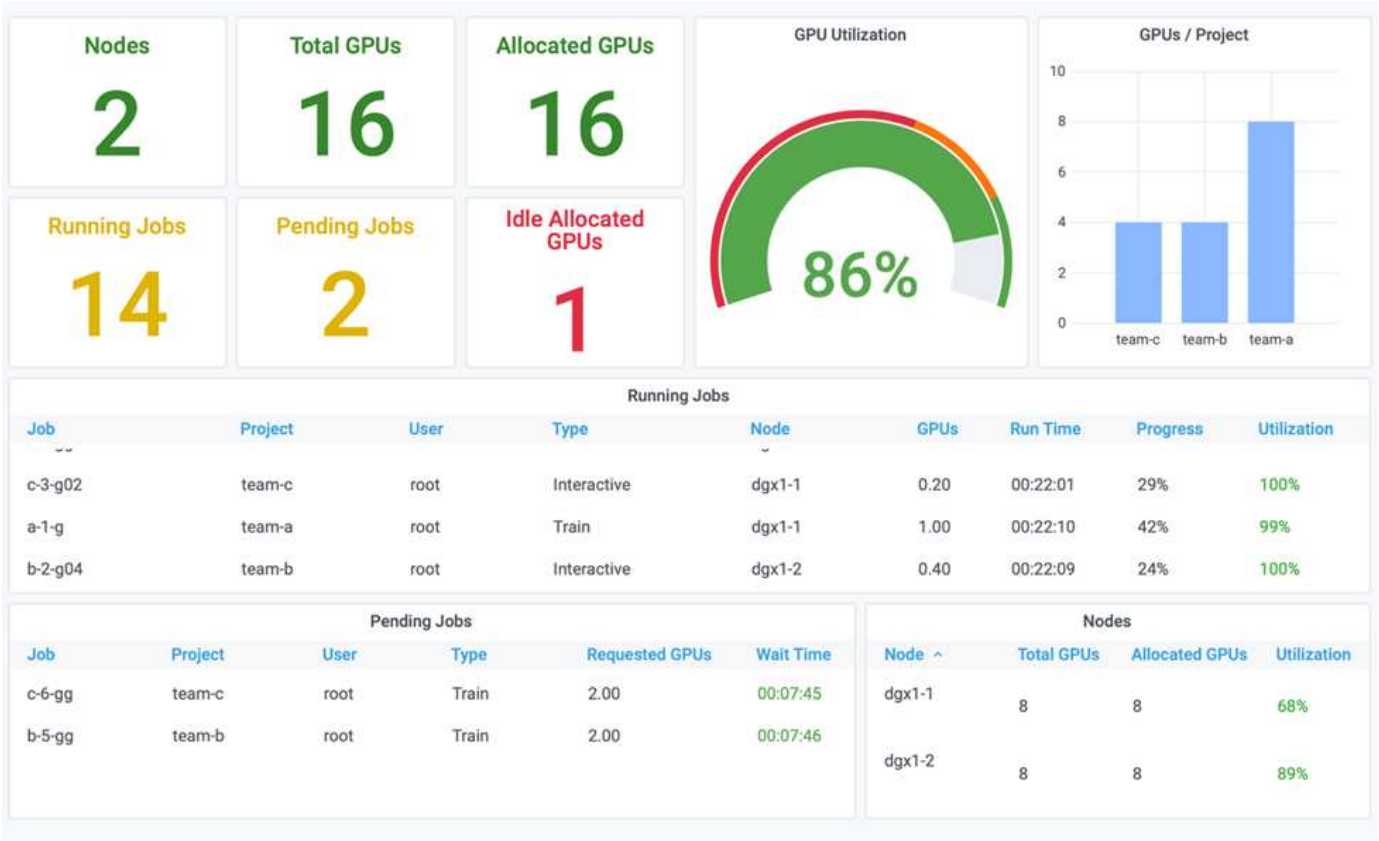
- 當多個團隊要求GPU超出配額時、顯示佇列機制。
- 示範系統如何根據配額之間的比率、在多個超出配額的團隊之間分配公平的叢集共用區、讓具有較大配額的團隊獲得較大的備用容量份額。

結束時 "基本資源配置公平性"有兩個工作負載排入佇列：一個用於「team b」、一個用於「team c」。在本節中、我們會排入其他工作負載的佇列。

如需工作提交、使用的容器映像及執行命令順序等詳細資料、請參閱 "第4.10節的測試詳細資料"。

當所有工作都根據區段提交時 "第4.10節的測試詳細資料"系統儀表板顯示：“team—a（團隊A）”、“team—b（團隊b）”和“team—c（團隊c）”都比預設配額多出GPU。與預設的軟配額（4個）相比、「團隊A」（team A）佔據4個GPU、而「團隊b」（team b）和「團隊c」（team c）則各佔2個GPU、而非軟配額（2個）。分配的配額過多GPU比例等於預設配額的比例。這是因為當多個團隊要求更多GPU、超過配額時、系統會使用預設配

額作為優先順序的參考資料、並據此進行資源配置。當企業資料科學團隊積極參與AI模式的開發與正式作業時、這種自動負載平衡可提供公平性和優先順序。



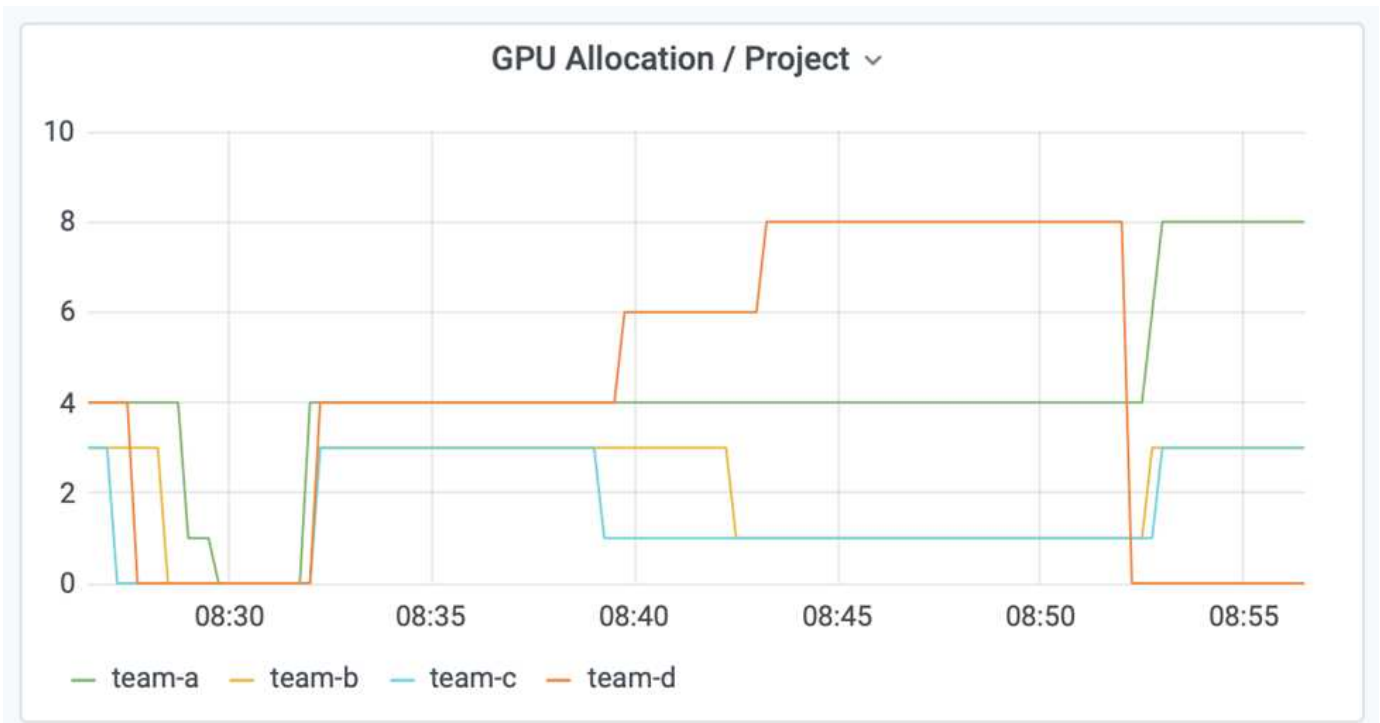
此測試案例的結果顯示下列項目：

- 系統開始將其他團隊的工作負載取消佇列。
- 根據公平演算法決定取消佇列的順序、例如「team-b」和「team-c」獲得相同數量的配額GPU（因為配額相似）、而A組的配額比B組和C組的配額高出兩倍、因此獲得兩倍的GPU數量。
- 所有分配都會自動完成。

因此、系統應穩定在下列狀態：

專案	已分配的GPU	留言
團隊A	8/4.	超過配額的四個GPU。空佇列。
團隊b	4/2.	超過配額的兩個GPU。一個工作負載已排入佇列。
團隊	4/2.	超過配額的兩個GPU。一個工作負載已排入佇列。
團隊	0/8.	完全不使用GPU、沒有排入佇列的工作負載。

下圖顯示各區段的「執行：AI分析」儀表板中、隨著時間推移、每個專案的GPU配置 "透過過度配額GPU配置、實現高叢集使用率"、"基本資源配置公平性"和 "過度配額的公平性"。圖中的每一行都會指出任何時間為特定資料科學團隊配置的GPU數量。我們可以看到、系統會根據提交的工作負載動態配置GPU。如此一來、當叢集中有可用的GPU時、團隊就能跳過配額、然後根據公平原則預先部署工作、最後才會達到四個團隊的穩定狀態。



將資料儲存至Trident佈建的PersistentVolume

NetApp Trident是完全受支援的開放原始碼專案、旨在協助您滿足容器化應用程式的複雜持續需求。您可以將資料讀寫至Trident佈建的Kubernetes PersistentVolume (PV)、並享有NetApp ONTAP 供應的資料分層、加密、NetApp Snapshot技術、法規遵循及高效能等額外優點。

重新使用現有命名空間中的PVCS

對於較大型的AI專案、不同的容器可能會更有效率地讀取資料並將資料寫入相同的Kubernetes PV。若要重複使用Kubernetes持續Volume宣告 (PVC)、使用者必須已建立一個PVC。請參閱 ["NetApp Trident文件"](#) 以取得建立永久虛擬基礎資料的詳細資料。以下是重複使用現有的永久虛擬資料的範例：

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

執行下列命令、查看專案「團隊A」的「PVC-TEST」工作狀態：

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

您應該會看到PV /tmp/pvc1mount掛載到「team A」工作「PVC-test」。如此一來、多個容器就能從同一個磁碟區讀取、當開發或正式作業中有多個相競的模型時、此功能非常實用。資料科學家可以建立一組模型、然後以多數投票或其他技術來結合預測結果。

使用下列項目存取Container Shell：

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

然後、您可以檢查掛載的Volume、並存取容器內的資料。

這項可重複使用PVCS的功能可搭配NetApp FlexVol 的功能、以及NetApp ONTAP FlexGroup 的NetApp功能、讓資料工程師擁有更靈活、更健全的資料管理選項、以充分運用NetApp技術所提供的資料架構。

結論

NetApp與Run：AI已在本技術報告中合作、展示NetApp ONTAP AI解決方案的獨特功能、並搭配Run:AI平台、以簡化AI工作負載的協調作業。上述步驟提供參考架構、可簡化資料管線和工作負載協調流程、以利深度學習。我們鼓勵想要實作這些解決方案的客戶聯絡NetApp、並聯絡Run：AI以取得更多資訊。

第4.8節的測試詳細資料

本節包含本節的測試詳細資料 ["透過過度配額GPU配置、實現高叢集使用率"](#)。

依下列順序提交工作：

專案	映像	# GPU	總計	留言
團隊A	Jupyter	1.	1/4	—
團隊A	NetApp	1.	2/4.	—
團隊A	執行：AI	2.	4/4.	使用所有配額
團隊b	執行：AI	0.6	0.6/2.	部分GPU
團隊b	執行：AI	0.4	1/2	部分GPU
團隊b	NetApp	1.	2/2.	—
團隊b	NetApp	2.	4/2.	超過配額2個
團隊	執行：AI	0.5	0.5/2.	部分GPU
團隊	執行：AI	0.3	0.8/2.	部分GPU
團隊	執行：AI	0.2	1/2	部分GPU
團隊	NetApp	2.	3/2.	一個超出配額
團隊	NetApp	1.	4/2.	超過配額2個
團隊	NetApp	4.	4/8.	使用一半配額

命令結構：

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

測試中使用的實際命令順序：

```

$ 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

```

此時、您應具備下列狀態：

專案	已分配的GPU	已排入佇列的工作負載
團隊A	4/4（軟配額/實際分配）	無
團隊b	4/2.	無
團隊	4/2.	無
團隊	4/8.	無

請參閱一節 ["透過過度配置GPU來實現高叢集使用率"](#) 以討論後續的測試案例。

第4.9節的測試詳細資料

本節包含本節的測試詳細資料 ["基本資源配置公平性"](#)。

依下列順序提交工作：

專案	# GPU	總計	留言
團隊	2.	6/8.	team b/c/ c工作負載暫停並移至「待處理」。
團隊	2.	8/8.	其他團隊（b / c）工作負載暫停、移至「待處理」。

請參閱下列執行的命令順序：

```
$ 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
```

此時、您應具備下列狀態：

專案	已分配的GPU	已排入佇列的工作負載
團隊A	4/4.	無
團隊b	2/2.	無
團隊	2/2.	無
團隊	8/8.	無

請參閱一節 "[基本資源配置公平性](#)" 以討論後續測試案例。

第4.10節的測試詳細資料

本節包含本節的測試詳細資料 "[過度配額的公平性](#)"。

請依下列順序提交工作：「team a」、「team-b」和「team-c」：

專案	# GPU	總計	留言
團隊A	2.	4/4.	1個工作負載排入佇列
團隊A	2.	4/4.	2個工作負載排入佇列
團隊b	2.	2/2.	2個工作負載排入佇列
團隊	2.	2/2.	2個工作負載排入佇列

請參閱下列執行的命令順序：

```
$ 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
```

此時、您應具備下列狀態：

專案	已分配的GPU	已排入佇列的工作負載
團隊A	4/4.	兩個工作負載分別要求GPU
團隊b	2/2.	兩個工作負載分別要求兩個GPU
團隊	2/2.	兩個工作負載分別要求兩個GPU

專案	已分配的 GPU	已排入佇列的工作負載
團隊	8/8.	無

接下來、刪除「team d」的所有工作負載：

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

請參閱一節 "[過度配額的公平性](#)"、以討論後續的測試案例。

何處可找到其他資訊

若要深入瞭解本文所述資訊、請參閱下列資源：

- NVIDIA DGX系統
 - NVIDIA DGX-1系統<https://www.nvidia.com/en-us/data-center/dgx-1/>
 - NVIDIA V100 Tensor Core GPU<https://www.nvidia.com/en-us/data-center/tesla-v100/>
 - NVIDIA NGC<https://www.nvidia.com/en-us/gpu-cloud/>
- RUN：AI Container協調解決方案
 - RUN：AI產品簡介<https://docs.run.ai/home/components/>
 - 執行：AI安裝文件<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 CLI中提交工作<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 CLI中分配GPU分數<https://docs.run.ai/Researcher/Walkthroughs/Walkthrough-Using-GPU-Fractions/>
- NetApp AI Control Plane
 - 技術報告<https://www.netapp.com/us/media/tr-4798.pdf>
 - 簡短示範https://youtu.be/gfr_sO27Rvo
 - GitHub儲存庫https://github.com/NetApp/kubeflow_jupyter_pipeline
- NetApp AFF 產品
 - NetApp AFF A系列產品型錄<https://www.netapp.com/us/media/ds-3582.pdf>
 - 適用於All Flash FAS 的NetApp Flash優勢<https://www.netapp.com/us/media/ds-3733.pdf>
 - 資訊庫ONTAP<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>
 - NetApp ONTAP FlexGroup 《銷售區》技術報告<https://www.netapp.com/us/media/tr-4557.pdf>
- NetApp ONTAP AI
 - 《採用DGX-1與Cisco Networking設計指南的AI》ONTAP<https://www.netapp.com/us/media/nva-1121-design.pdf>

- 《採用DGX-1和Cisco Networking部署指南的AI》ONTAP<https://www.netapp.com/us/media/nva-1121-deploy.pdf>
- 《採用DGX-1和Mellanox網路設計指南的AI》ONTAP<http://www.netapp.com/us/media/nva-1138-design.pdf>
- 《採用DGX-2設計指南的AI》ONTAP<https://www.netapp.com/us/media/nva-1135-design.pdf>

TR-4799-Design：適用於ONTAP 自動駕駛工作負載的NetApp AI參考架構

NetApp公司David Arette和Sung-HAN LIN

NVIDIA DGX系統系列是全球第一款專為企業AI打造的整合式人工智慧（AI）平台。NetApp AFF 供應極致效能與領先業界的混合雲資料管理功能。NetApp與NVIDIA聯手打造NetApp ONTAP 的「可靠性AI」參考架構、為客戶提供一套立即可用的解決方案、以企業級效能、可靠性與支援來支援AI與機器學習（ML）工作負載。

"TR-4799-Design：適用於ONTAP 自動駕駛工作負載的NetApp AI參考架構"

TR-4811：適用於ONTAP 醫療的NetApp SAI參考架構：診斷影像-解決方案設計

Rick Huang、Sung-HAN LIN、Sathish Thyagarajan、NetApp JacCI Cenci、NVIDIA

這項參考架構提供指引、協助客戶使用NVIDIA DGX-2系統和NetApp AFF 支援醫療使用案例的NetApp支援中心、打造人工智慧（AI）基礎架構。其中包含開發深度學習（DL）模型、用於醫療診斷影像、驗證測試案例及結果的高層工作流程相關資訊。其中也包括客戶部署的規模調整建議。

"TR-4811：適用於ONTAP 醫療的NetApp SAI參考架構：診斷影像-解決方案設計"

TR-4807：ONTAP 適用於金融服務工作負載的NetApp AI參考架構-解決方案設計

Karthikeyan Nagalingam、Sung-HAN LIN、NetApp JacCI Cenci、NVIDIA

此參考架構為使用NVIDIA DGX-1系統和NetApp AFF 支援財務部門使用案例的NetApp支援中心來建置人工智慧基礎架構的客戶提供準則。其中包含開發金融服務測試案例與結果深度學習模式所用的高層工作流程相關資訊。其中也包括客戶部署的規模調整建議。

"TR-4807：ONTAP 適用於金融服務工作負載的NetApp AI參考架構-解決方案設計"

推廣 AI 和 NetApp 價值

作者：Sathish Thyagarajan、NetApp

摘要

對泛用人工智慧（AI）的需求、正推動產業的顛覆、增強商業創造力和產品創新。許多組織都在使用泛型 AI 來建置新產品功能、改善工程生產力、並以 AI 驅動的應用程式為原型、以提供更好的結果和消費者體驗。泛用 AI（例如推廣學前訓練的 Transformer（GPT））使用神經網路來建立新內容、例如文字、音訊和視訊等。鑑於大型語言模型（LLM）涉及極大規模的資料集、因此架構強大的 AI 基礎架構至關重要、因為它充分利用內部部署、混合式和多雲端部署選項的強大資料儲存功能、並降低與資料移動性相關的風險、資料保護與治理、讓公司能夠先設計 AI 解決方案。本白皮書說明這些考量事項及對應的 NetApp® AI 功能、可在 AI 資料傳輸管道中順暢地進行資料管理及資料移動、以進行訓練、再訓練、微調及推斷一般 AI 模型。

執行摘要

最近在 2022 年 11 月 GPT-3 推出的 ChatGPT 之後、針對使用者提示而產生文字、程式碼、影像、甚至是治療性蛋白質的新 AI 工具已獲得顯著的名聲。這表示使用者可以使用自然語言提出要求、AI 將會解譯並產生文字、例如新聞文章或產品說明、這些文章或產品說明會反映使用者的要求、或是使用已訓練的現有資料演算法來產生程式碼、音樂、語音、視覺效果和 3D 資產。因此、在 AI 系統的設計中、像是穩定擴散、幻覺、快速工程和價值調整等詞句、都在迅速出現。這些自我監督或半監督的機器學習（ML）模式正透過雲端服務供應商和其他 AI 公司廠商、廣泛以預先訓練的基礎模式（FM）提供使用、這些模式正被各行各業的各種商業機構採用、用於各種下游的 NLP（自然語言處理）工作。如 McKinsey 等研究分析公司所宣稱：「通用 AI 對生產力的影響、可能會為全球經濟帶來數億美元的價值。」雖然公司正在重新想像 AI 是人類的思想合作夥伴、而 FMS 也在同時拓展業務和機構對於泛型 AI 的能力、但管理大量資料的機會仍會持續成長。本文件提供有關泛用 AI 的簡介資訊、以及與 NetApp 功能相關的設計概念、這些功能可為 NetApp 客戶帶來內部部署和混合式或多雲端環境的價值。

因此、客戶在 AI 環境中使用 NetApp 的 IT 環境有哪些？* NetApp 協助企業組織因快速資料與雲端成長、多雲端管理、以及採用新一代技術（例如 AI）而產生複雜度。NetApp 將各種功能整合到智慧型資料管理軟體和儲存基礎架構中、這些功能與針對 AI 工作負載最佳化的高效能相平衡。LLM 等泛型 AI 解決方案需要多次從儲存設備讀取及處理來源資料集、以加強智慧。NetApp 在邊緣對核心對雲端生態系統的資料移動性、資料治理和資料安全技術方面、一直是領導者、為企業客戶建置大規模 AI 解決方案提供服務。NetApp 擁有強大的合作夥伴網路、協助資料長、AI 工程師、企業架構設計師和資料科學家設計可自由流動的資料傳輸管道、以供資料準備、資料保護、以及 AI 模型訓練與推斷的策略性資料管理責任、將 AI/ML 生命週期的效能與擴充性最佳化。NetApp 資料技術與功能、例如用於深度學習資料傳輸管道的 NetApp® ONTAP AI®、用於在儲存端點之間無縫且有效率地傳輸資料的 NetApp® SnapMirror®、NetApp® FlexCache® 可在資料流從批次轉移至即時時即時呈現、而資料工程則會在迅速完成時進行、為部署即時通用 AI 模型帶來價值。隨著各種類型的企業紛紛採用新的 AI 工具、他們將面臨資料挑戰、從邊緣到資料中心、再到需要可擴充、負責且可解釋的 AI 解決方案的雲端。身為混合雲和多雲端的資料管理機構、NetApp 致力於打造合作夥伴和聯合解決方案網路、協助建構資料管線和資料湖的所有層面、以便進行泛用 AI 模型訓練（訓練前）、微調、情境式推斷和模式衰退監控 LLM。









什麼是泛型 AI？

泛用 AI 正在改變我們建立內容的方式、產生新的設計概念、以及探索新穎的組合。它說明了神經網路架構、像是泛型對抗網路（GAN）、變型自動編碼器（VAE）、以及泛型預訓練的變形器（GPT）、這些架構可產生新的內容、例如文字、程式碼、影像、音訊、視訊、和綜合資料。OpenAI 的 Chat-GPT、Google 的 Bard、擁抱樺面布盧姆和 Meta 的 llama 等以變壓器為基礎的模型、已成為大型語言模型許多進步的基礎技術。同樣地、OpenAI 的 Dall-E、Meta 的 CM3leon 和 Google 的 Imagen 都是文字對影像擴散模型的範例、為客戶提供前所未有的寫實度、可從頭建立新的複雜影像、或是編輯現有影像、以使用資料集擴充和文字對影像合成、連結文字和視覺義學、產生高品質的內容感知影像。數位藝術家開始結合使用 nerf（神經放射照野）等描繪技術與泛變 AI、將靜態 2D 影像轉換成身歷其境的 3D 場景。一般而言、LLM 有四個參數：（1）模型大小（通常以十億個參數為單位）；（2）訓練資料集的大小；（3）訓練成本；（4）訓練後的示範效能。LLM 也主要屬於三種變壓器架構。（i）僅編碼器機型。例如 Bert（Google、2018）；（ii）編碼解碼器、例如 BART（Meta、2020）和（iii）純解碼器機型。例如：llama（Meta、2023）、Palm-E（Google、2023）。根據業務需求、無論公司選擇的架構為何、訓練資料集中的模型參數（N）數量和權杖（D）數量、通常都會決定訓練（訓練前）或微調 LLM 的基礎成本。

企業使用案例和下游 NLP 工作

各行各業的企業都發現、AI 從現有資料中擷取並產生新形式的價值、以用於商業營運、銷售、行銷和法律服務。根據 IDC（International Data Corporation）關於全球通用 AI 使用案例和投資的市場情報、軟體開發和產品設計的知識管理將受到最大的影響、其次是建立行銷腳本和開發人員程式碼。在醫療領域、臨床研究組織正在開創新的醫療領域。經過預先訓練的模式、例如 ProteinBERT、將基因學（GO）標註納入醫療藥物的蛋白質結構快速設計、這是藥物探索、生物資訊學和分子生物學的重大里程碑。Biotech 公司已開始進行人工智慧型發現醫學的人類試驗、其目的是治療肺部纖維化（IPF）等疾病、這種肺部疾病會造成無法還原的肺部組織創傷。

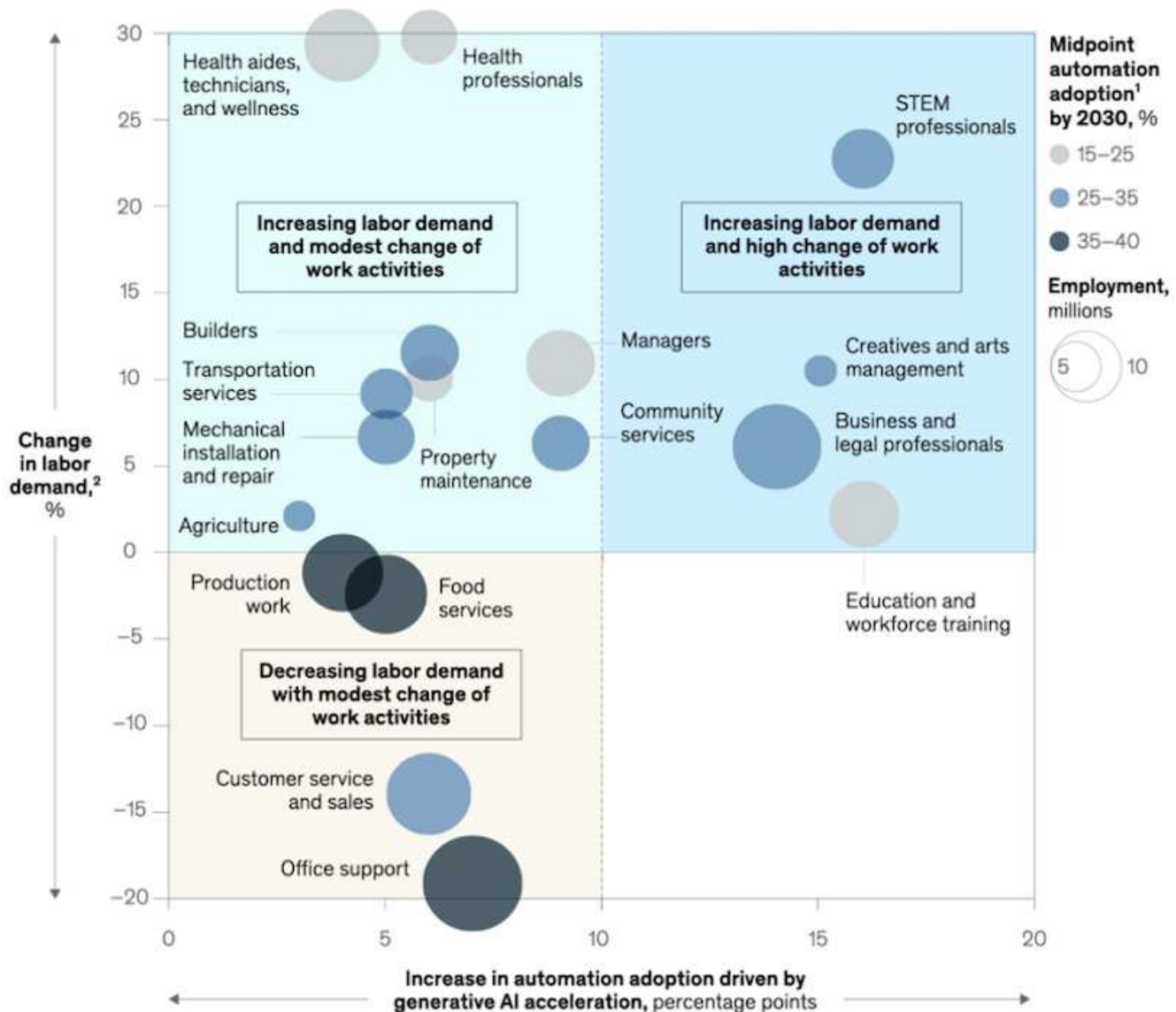
圖 1：使用案例來推動泛用 AI

 Chatbots	 Drug discovery
 Text generation	 Genome model expression
 Image generation	 Classification
 Code generation	 Speech-to-Text

由泛用 AI 帶動的自動化採用率增加、也改變了許多職業的工作活動供應與需求。根據 McKinsey 的資料、美國勞動力市場（下圖）經歷了快速的轉型、只有在考量 AI 的影響時、才能持續轉型。

資料來源：McKinsey & Company

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



儲存在泛用 AI 中的角色

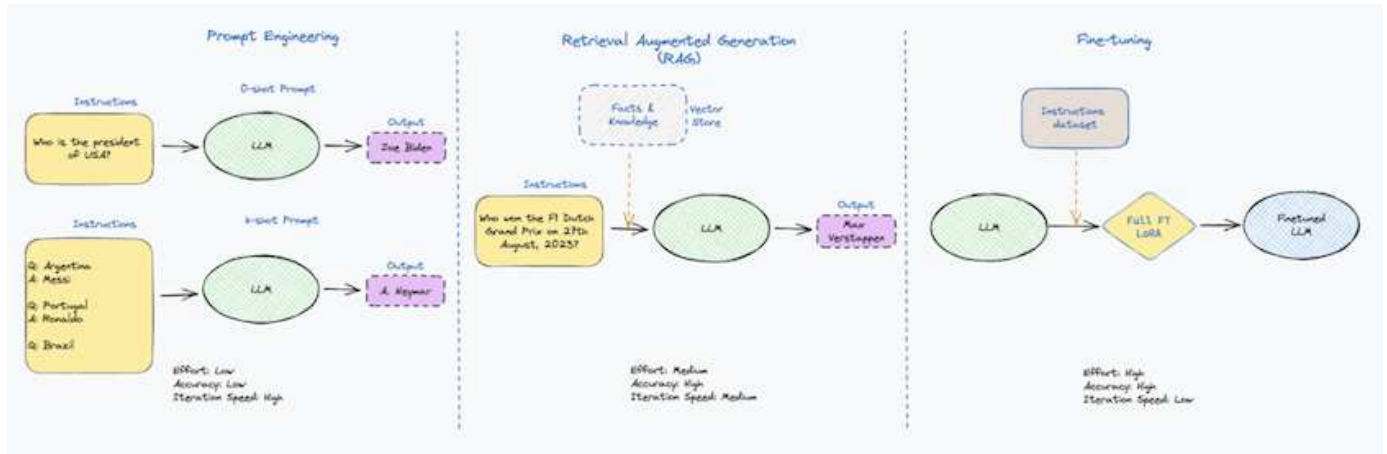
LMS 主要仰賴深度學習、GPU 和運算。不過、GPU 緩衝區填滿時、資料必須快速寫入儲存設備。雖然某些 AI 機型的規模足夠小、足以在記憶體中執行、但 LMS 需要高 IOPS 和高處理量儲存設備、才能快速存取大型資料集、尤其是當它涉及數十億個權杖或數百萬個影像時。對於 LLM 的一般 GPU 記憶體需求、以 10 億個參數訓練模型所需的記憶體、最高可達 80 GB @32 位元的完整精確度。在這種情況下、Meta's Llama 2 是規模從 70 億到 700 億參數不等的 LMS 系列、可能需要 70x80、約 5600 GB 或 5.6TB 的 GPU RAM。此外、您所需的記憶體容量與您想要產生的最大權杖數量直接成比例。例如、如果您想要產生最多 512 個權杖（約 380 個字）的輸出、則需要 "512MB"。這似乎並不重要、但如果您想要執行較大的批次、就會開始增加。因此、組織在記憶體中訓練或微調 LMS 的成本非常昂貴、因此儲存設備成為泛用 AI 的基石。

三種主要的 LRAM 方法

對於大多數企業而言、根據目前的趨勢、部署 LDM 的方法可以精簡為 3 個基本案例。如最近所述 "《哈佛商業評論》" 文章：（1）從頭開始訓練（課前訓練）LLM – 成本高昂、需要專業的 AI/ML 技能；（2）使用企業資料微調基礎模型 – 複雜但可行；（3）使用擷取擴充產生（RAG）來查詢包含公司資料的文件儲存庫、API 和向量資料庫。每個解決方案都會在其實作中的工作、迭代速度、成本效益和模型準確度之間取得平衡、用於解決

不同類型的問題（下圖）。

圖 3：問題類型



基礎模式

基礎模型（FM）也稱為基礎模型、是一種大型 AI 模型（LLM）、針對大量未經標記的資料進行訓練、大規模使用自我監督、通常適用於各種下游 NLP 工作。由於訓練資料並未由人類標示、因此模型會出現、而非明確編碼。這表示模型可以產生自己的故事或敘述、而無需明確設定。因此 FM 的重要特徵是同質化、這表示許多網域都使用相同的方法。不過、透過個人化和微調技術、FMS 整合到現在出現的產品中、不僅能產生文字、文字對影像和文字對程式碼、也能說明網域特定的工作或偵錯程式碼。例如、像 OpenAI 的 Codex 或 Meta 的 Code Llama 等 FMS、可根據程式設計工作的自然語言說明、以多種程式設計語言產生程式碼。這些模型精通十多種程式設計語言、包括 Python、C#、JavaScript、Perl、Ruby、和 SQL。他們瞭解使用者的意圖、並產生特定的程式碼、以完成所需的工作、對軟體開發、程式碼最佳化及程式設計工作自動化非常有用。

微調、網域特性和重新訓練

在資料準備和資料預先處理之後、LLM 部署的常見做法之一是選擇經過預先訓練的模型、該模型已在大型且多元化的資料集上接受訓練。在微調的情況下、這可以是開放原始碼的大型語言模型、例如 "中繼的 Llama 2." 針對 700 億個參數和 2 兆個記號進行訓練。選擇預先訓練的模型後、下一步是針對特定網域的資料進行微調。這包括調整模型的參數、並訓練其新資料、以因應特定的網域和工作。例如、BloombergGPT 是專屬 LLM、專門針對金融業的各種財務資料進行訓練。專為特定工作所設計和訓練的網域特定模型、通常在其範圍內具有較高的準確度和效能、但在其他工作或網域之間的傳輸能力較低。當業務環境和資料在一段時間內變更時、相較於測試期間的效能、FM 的預測準確度可能會開始下降。這是重新訓練或微調模型的關鍵所在。傳統 AI / ML 的模型重新訓練是指使用新資料更新已部署的 ML 模型、通常是為了消除發生的兩種類型的資料外包。（1）概念漂移—當輸入變數與目標變數之間的連結隨著時間而改變時、由於我們要預測變更的內容說明、因此模型可能會產生不正確的預測。（2）資料漂移：當輸入資料的特性發生變化時、例如隨著時間而改變客戶習慣或行為、因此模型無法回應此類變更。同樣地、重新訓練也適用於 FMS/LLMs、不過可能會花費大量成本（以百萬美元計）、因此大多數組織可能不會考慮這樣做。它正在積極研究中、仍在 LLM Ops 領域中出現。因此、當精細調校的 FMS 發生模式衰退時、企業可能會選擇使用較新的資料集再次微調（較便宜）、而非重新訓練。從成本觀點來看、以下是 Azure OpenAI Services 的模型價格表範例。對於每個工作類別、客戶可以微調和評估特定資料集上的模型。

資料來源：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

迅速進行工程與推斷

「提示工程」是指如何與 LLM 通訊、以執行所需工作而不更新模型權重的有效方法。如同 AI 模型訓練和微調對 NLP 應用程式一樣重要、推斷也同樣重要、訓練模型會回應使用者提示。推斷的系統需求通常更多地說明 AI 儲存系統的讀取效能、因為 AI 儲存系統需要套用數十億個儲存的模型參數、才能產生最佳回應、因此會將資料從 LMS 饋送至 GPU 。

LLMOps 、 Model Monitoring 和 VectorStores

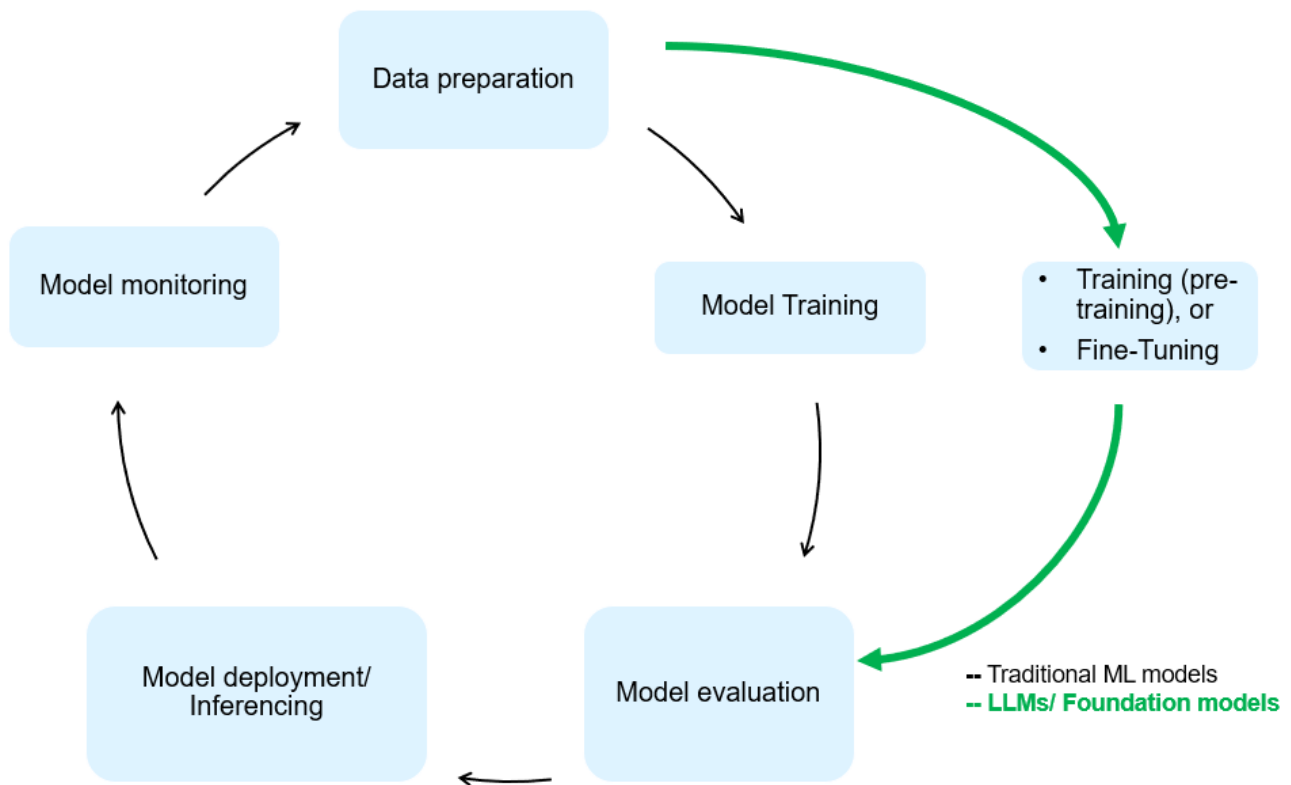
與傳統機器學習作業（MLOps）一樣、大型語言模型作業（LLMOps）也需要資料科學家和 DevOps 工程師合作、提供工具和最佳實務、以便在正式作業環境中管理 LMS。然而、LLM 的工作流程和技術堆疊在某些方面可能會有所不同。例如、LLM 管線是使用 LangChain 字串等架構建置、將多個 LLM API 呼叫結合到外部嵌入端點、例如 vectorStores 或向量資料庫。使用嵌入式端點和 Vectorstore 作為下游連接器（例如向量資料庫）、代表資料儲存和存取方式的重大發展。相較於從頭開始開發的傳統 ML 模型、LMS 通常仰賴傳輸學習、因為這些模型是從 FMS 開始、而 FMS 則是以新資料進行微調、以改善更特定領域的效能。因此、LLMOps 提供風險管理和模式衰退監控的功能至關重要。

泛用 AI 時代的風險與道德規範

「ChatGPT – 它很流暢、但仍是一大堆廢話。」 – MIT 技術評論。垃圾回收、總是運算上的難題。與泛用 AI 的唯一差異在於、它在讓垃圾變得高度可信、導致結果不準確方面表現優異。LMS 很容易發明事實、以符合其建置的敘述。因此、將泛用 AI 視為降低成本的大好機會的公司、必須有效偵測深層假貨、減少偏見、降低風險、以保持系統誠實且合乎道德。具備強大 AI 基礎架構的自由流動資料傳輸管道、可透過端對端加密支援資料移動性、資料品質、資料治理和資料保護、而 AI guardrails 則在設計負責任且可解釋的 AI 模型時表現出色。

客戶案例和 NetApp

圖 3：機器學習 / 大型語言模型工作流程



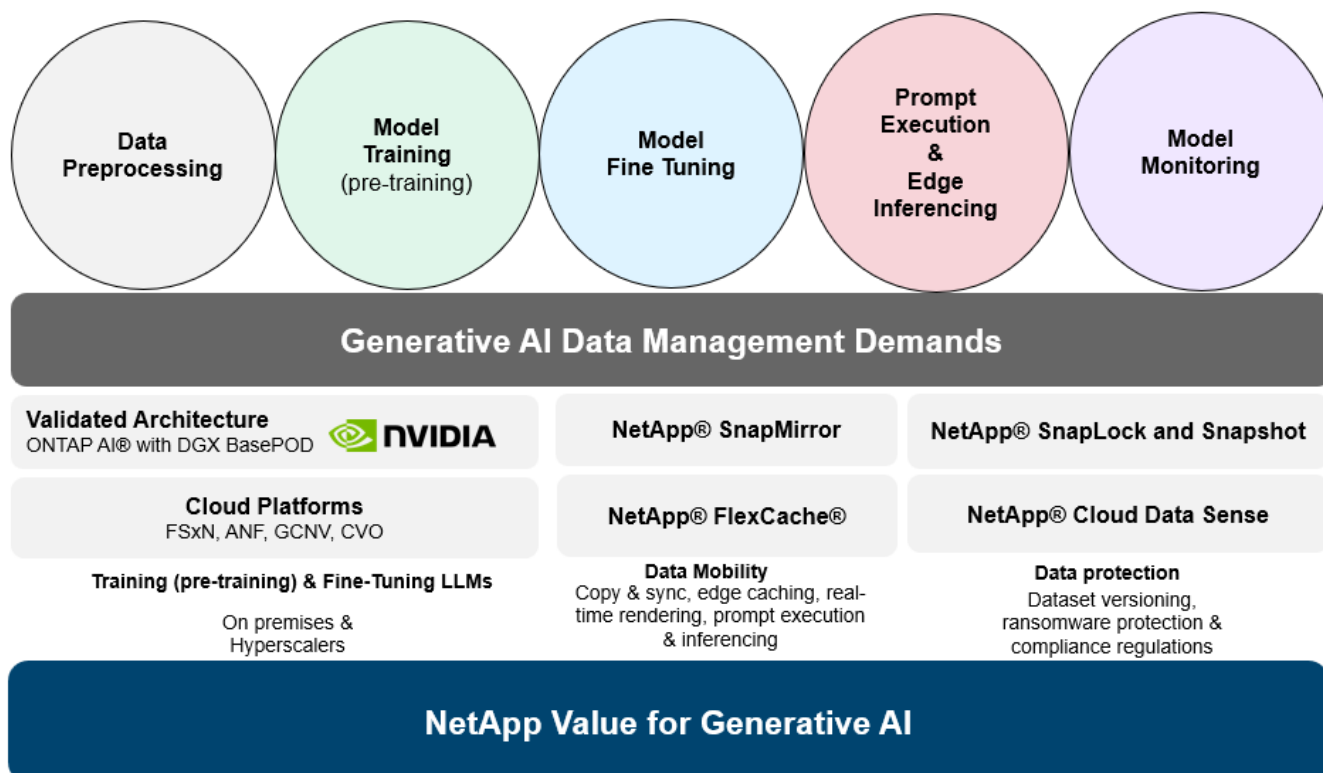
- 我們是否接受訓練或微調？ * 是否（a）從頭開始訓練 LLM 模型、微調預先訓練的 FM、或使用 RAG 從基礎模型以外的文件儲存庫擷取資料、並加強提示、以及（b）利用開放原始碼 LMS（例如 Llama 2）或專屬 FMS（例如 ChatGPT、Bard、AWS bedrock）、都是組織的策略決策。每種方法都會在成本效益、資料嚴重性、營運、模型準確度和 LLMS 管理之間取得平衡。

NetApp 身為一家公司、在其工作文化中、以及在產品設計和工程工作的方法中、均採用 AI。例如、NetApp 的自動勒索軟體保護是使用 AI 和機器學習技術來建置。它能及早偵測檔案系統異常狀況、協助在威脅影響作業之前先識別威脅。其次、NetApp 使用預測性 AI 進行業務營運、例如銷售和庫存預測及聊天機器人程式、協助客戶取得客服中心產品支援服務、技術規格、保固、服務手冊等。第三、NetApp 透過產品與解決方案為客戶打造預測性 AI 解決方案、例如需求預測、醫療影像、情緒分析、此外、Gans 等泛用 AI 解決方案可在製造業界偵測工業影像異常狀況、以及利用 NetApp® ONTAP AI®、NetApp® SnapMirror® 和 NetApp® FlexCache® 等 NetApp 產品與功能、在銀行與金融服務中進行反洗錢和詐騙偵測。

NetApp 功能

在一般 AI 應用程式（例如 chatbot、程式碼產生、影像產生或基因組模型表達式）中、資料的移動和管理可跨越邊緣、私有資料中心和混合式多雲端生態系統。例如、透過預先訓練模型（例如 ChatGPT）的 API、協助乘客從終端使用者應用程式將機票升級至商務艙的即時 AI 機器人、無法自行完成這項工作、因為無法在網際網路上公開乘客資訊。API 需要從航空公司取得乘客的個人資訊和機票資訊、這些資訊可能存在於混合式或多雲端生態系統中。類似的案例可能適用於透過終端使用者應用程式分享藥物分子和病患資料的科學家、該應用程式使用 LMS 在涉及一對多生物醫療研究機構的藥物探索中完成臨床試驗。傳遞給 FMS 或 LRAM 的敏感資料可能包括 PII、財務資訊、健全狀況資訊、生物特徵資料、位置資料、通訊資料、線上行為和法律資訊。在即時呈現、快速執行和邊緣推斷的情況下、資料會透過開放原始碼或專屬 LLM 模式、從終端使用者應用程式移至儲存端點、移至內部部署或公有雲平台上的資料中心。在所有這類案例中、資料移動性和資料保護對於仰賴大型訓練資料集和移動此類資料的 LLMS 所涉及的 AI 作業而言、是至關重要的。

圖 4：泛用 AI - LLM 資料管道



NetApp 的儲存基礎架構、資料和雲端服務產品組合均採用智慧型資料管理軟體。

- 資料準備 *：LLM 技術堆疊的第一個支柱基本上未受舊版傳統 ML 堆疊的影響。在訓練或微調之前、必須先在 AI 管道中預先處理資料、才能將資料標準化及清除。此步驟包括以 Amazon S3 層或內部部署儲存系統（例如檔案存放區或 NetApp StorageGRID 等物件存放區）形式、在任何位置擷取資料的連接器。
- NetApp® ONTAP® 是一項基礎技術、是 NetApp 在資料中心和雲端的關鍵儲存解決方案的基礎。ONTAP 包含各種資料管理與保護功能、包括自動勒索軟體防護、防範網路攻擊、內建資料傳輸功能、以及儲存效率功能、適用於各種架構、從內部部署、混合式、NAS、SAN、物件、以及 LLM 部署的軟體定義儲存（SDS）情況。
- NetApp® ONTAP AI® 提供深度學習模式訓練。NetApp® ONTAP® 支援 NVIDIA GPU Direct Storage™、並使用 NFS over RDMA、為具有 ONTAP 儲存叢集和 NVIDIA DGX 運算節點的 NetApp 客戶提供支援。它提供極具成本效益的效能、可從儲存設備讀取及處理大量來源資料集至記憶體、藉此強化智慧功能、讓組織能夠接受訓練、微調及擴充對 LMS 的存取。
- NetApp® FlexCache® 是一種遠端快取功能、可簡化檔案發佈、並僅快取主動讀取的資料。這對於 LLM 訓練、重新訓練和微調都很有用、可為具有即時呈現和 LLM 推斷等業務需求的客戶帶來價值。
- NetApp® SnapMirror® 是一項 ONTAP 功能、可在任何兩個 ONTAP 系統之間複寫磁碟區快照。此功能可將邊緣的資料最佳傳輸至內部部署資料中心或雲端。當客戶想要在雲端中開發包含企業資料的磁碟機時、SnapMirror 可用於在內部部署和超大規模雲端之間安全且有效率地移動資料。它只能有效傳輸變更、節省頻寬並加速複寫、因此在 FMS 或 LMS 的訓練、重新訓練和微調作業期間、能提供必要的資料移動功能。
- NetApp® SnapLock® 可在 ONTAP 型儲存系統上提供不變的磁碟功能、以供資料集版本管理。微核心架構是專為使用 FPolicy™ Zero Trust 引擎保護客戶資料而設計。當攻擊者以特別耗用資源的方式與 LLM 互動時、NetApp 可抵抗拒絕服務（DoS）攻擊、確保客戶資料可用。
- NetApp® Cloud Data Sense® 可協助識別、對應及分類企業資料集中的個人資訊、制定原則、滿足內部部署或雲端的隱私權要求、協助改善安全狀態並遵守法規。
- NetApp® BlueXP™ 分類、採用 Cloud Data Sense 技術。客戶可以自動掃描、分析、分類資料、並對資料進行行動、偵測安全風險、最佳化儲存設備、並加速雲端部署。它透過統一化的控制平台結合儲存與資料

服務、客戶可以使用 GPU 執行個體來進行運算、並將混合式多雲端環境用於冷儲存分層、以及歸檔與備份。

- **NetApp 檔案物件雙重性 ***。NetApp ONTAP 可為 NFS 和 S3 啟用雙傳輸協定存取。有了這個解決方案、客戶就能透過 NetApp Cloud Volumes ONTAP 的 S3 儲存區、從 Amazon AWS SageMaker 筆記型電腦存取 NFS 資料。這可為需要輕鬆存取異質資料來源的客戶提供靈活彈性、並能從 NFS 和 S3 共用資料。例如、在 SageMaker 上微調 FMS、例如 Meta 的 Llama 2 文字產生模型、可存取檔案物件儲存區。
- **NetApp® Cloud Sync *** 服務提供簡單且安全的方法、可將資料移轉至雲端或內部部署的任何目標。Cloud Sync 可在內部部署或雲端儲存設備、NAS 和物件存放區之間順暢地傳輸和同步資料。
- **NetApp XCP*** 是一種用戶端軟體、可快速且可靠地進行任何對 NetApp 和 NetApp 對 NetApp 的資料移轉。XCP 也能將大量資料從 Hadoop HDFS 檔案系統有效移至 ONTAP NFS、S3 或 StorageGRID 及 XCP 檔案分析、提供檔案系統的可見度。
- **NetApp® DataOps Toolkit*** 是一種 Python 程式庫、可讓資料科學家、DevOps 和資料工程師輕鬆執行各種資料管理工作、例如近乎即時的資源配置、複製或快照資料 Volume 或 JupyterLab 工作區、並以高效能橫向擴充 NetApp 儲存設備作為後盾。
- **NetApp 的產品安全 ***。LMS 可能會在回應中不慎洩漏機密資料、因此對 CISO 表示疑慮、CISO 會研究利用 LRAM 的 AI 應用程式相關弱點。如 OWAASP（開放全球應用程式安全專案）所述、資料中毒、資料外洩、LMS 內的拒絕服務和即時注入等安全問題、可能會影響企業、使企業無法從資料外洩到未經授權的存取服務攻擊者。資料儲存需求應包括結構化、半結構化和非結構化資料的完整性檢查和不可變更的快照。NetApp 快照和 SnapLock 正用於資料集版本管理。它提供嚴格的角色型存取控制（RBAC）、以及安全的通訊協定、以及業界標準加密、可保護靜止和傳輸中的資料。Cloud Insights 和 Cloud Data Sense 共同提供各種功能、可協助您預先識別威脅的來源、並優先處理要還原的資料。

* ONTAP AI 搭配 DGX BasePOD *

NetApp® ONTAP® AI 參考架構搭配 NVIDIA DGX BasePOD、是適用於機器學習（ML）和人工智慧（AI）工作負載的可擴充架構。對於 LMS 的關鍵訓練階段、資料通常會定期從資料儲存設備複製到訓練叢集。此階段使用的伺服器使用 GPU 來平行化運算、為資料帶來極大的需求。滿足原始 I/O 頻寬需求、對於維持高 GPU 使用率至關重要。

* 採用 NVIDIA AI Enterprise* 的 ONTAP AI

NVIDIA AI Enterprise是一套端點對端點、雲端原生的AI與資料分析軟體套件、經過NVIDIA最佳化、認證及支援、可在採用NVIDIA認證系統的VMware vSphere上執行。此軟體可在現代化的混合雲環境中、輕鬆快速地部署、管理及擴充AI工作負載。NVIDIA AI Enterprise採用NetApp與VMware技術、以簡化且熟悉的套件提供企業級AI工作負載與資料管理功能。

* 單一雲端平台 *

完全託管的雲端儲存方案可在 Microsoft Azure 上原生提供、例如 Azure NetApp Files（anf）、AWS、Amazon FSx for NetApp ONTAP（FSxN）、以及 Google 雲端 NetApp Volumes（GNCV）。1P 是一套託管的高效能檔案系統、可讓客戶在公有雲中執行高可用度的 AI 工作負載、並改善資料安全性、以微調 LM/FMS 搭配 AWS SageMaker、Azure OpenAI Services 和 Google 的 Vertex AI 等雲端原生 ML 平台。

NetApp 合作夥伴解決方案套件

除了核心資料產品、技術和功能之外、NetApp 也與強大的 AI 合作夥伴網路密切合作、為客戶帶來更多附加價值。

- **AI 系統中的 NVIDIA guardrails *** 可作為保障措施、確保以合乎道德且負責任的方式使用 AI 技術。AI 開發人員可以選擇針對特定主題定義 LLM 應用程式的行為、並防止他們參與討論不必要的主題。Guardrails 是開

放原始碼工具組、可讓 LLM 與其他服務無縫且安全地連線、以建置值得信賴、安全且安全的 LLM 交談系統。

- Domino Data Lab* 提供多元化的企業級工具、無論您在 AI 旅程中的任何地方、都能快速、安全且經濟地建置及製作通用 AI。有了 Domino 的 Enterprise MLOps 平台、資料科學家就能使用偏好的工具及其所有資料、隨時隨地輕鬆訓練及部署模型、並從單一控制中心以具成本效益的方式管理風險。
- 適用於 Edge AI* 的 Modzy。NetApp® 與 Modzy 攜手合作、為任何類型的資料（包括影像、音訊、文字和表格）提供大規模的 AI。Modzy 是用於部署、整合及執行 AI 模型的 MLOps 平台、為資料科學家提供模型監控、漂移偵測及可探索性的功能、並提供整合式解決方案、以實現無縫 LLM 推斷。
- 執行：AI* 與 NetApp 合作、以 Run:AI 叢集管理平台來示範 NetApp ONTAP AI 解決方案的獨特功能、以簡化 AI 工作負載的協調。它會自動分割並加入 GPU 資源、透過 Spark、Ray、dask 和 Rapids 的內建整合架構、將資料處理管線擴充至數百部機器。

結論

只有在模型接受品質資料的大量訓練時、泛用 AI 才能產生有效的結果。雖然 LMS 已達成卓越的里程碑、但必須瞭解其與資料移動性和資料品質相關的限制、設計挑戰和風險。LMS 仰賴來自異質資料來源的大型和獨立訓練資料集。模型產生的不正確結果或不公正結果、可能會使企業和消費者面臨危險。這些風險可能會對應資料品質、資料安全性和資料移動性等資料管理挑戰所帶來的潛在 LLM 限制。NetApp 可協助企業組織因資料快速成長、資料移動性、多雲端管理及採用 AI 所帶來的複雜性。大規模的 AI 基礎架構和高效率的資料管理對於定義 AI 應用程式（例如泛用 AI）的成功與否至關重要。關鍵客戶涵蓋所有的部署情境、而不影響企業在控制成本效益、資料治理和道德 AI 實務的同時、隨需擴充的能力。NetApp 持續協助客戶簡化及加速 AI 部署。

TR-4785：採用NetApp E系列和BeeGFS的AI部署

Nagalakshmi raju、丹尼爾·蘭德斯、Nathan Swartz、Amine Bennani、NetApp

人工智慧（AI）、機器學習（ML）和深度學習（DL）應用程式涉及大型資料集和高運算能力。若要成功執行這些工作負載、您需要敏捷的基礎架構、讓您能夠順暢地橫向擴充儲存和運算節點。本報告包含在分散式模式中執行AI訓練模型的步驟、可無縫橫向擴充運算和儲存節點。報告也包含各種效能指標、說明如何將NetApp E系列儲存設備與BeeGFS平行檔案系統結合、為AI工作負載提供靈活、具成本效益且簡單的解決方案。

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

NVA-1150-Design：Quantum StorNext with NetApp E系列系統設計指南

Ryan Rodine、NetApp

本文件詳細說明如何使用NetApp E系列儲存系統來設計StorNext平行檔案系統解決方案。本解決方案涵蓋NetApp EF280 All Flash Array、NetApp EF300 All Flash NVMe Array、EF600 All Flash NVMe Array及NetApp E5760混合式系統。它提供以「最出色的基準測試」為基礎的效能特性分析、此工具廣泛用於媒體與娛樂產業的測試。

["NVA-1150-Design：Quantum StorNext with NetApp E系列系統設計指南"](#)

NVA-1150-Deploy：Quantum StorNext with NetApp E系列系統部署指南

Ryan Rodine、NetApp

本文件詳細說明如何使用NetApp E系列儲存系統部署StorNext平行檔案系統解決方案。本解決方案涵蓋NetApp EF280 All Flash Array、NetApp EF300 All Flash NVMe Array、NetApp EF600 All Flash NVMe Array及NetApp E5760混合式系統。它提供以「最出色的基準測試」為基礎的效能特性分析、此工具廣泛用於媒體與娛樂產業的測試。

["NVA-1150-Deploy：Quantum StorNext with NetApp E系列系統部署指南"](#)

版權資訊

Copyright © 2024 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。