



分散式**Azure**訓練：Lane偵測 NetApp Solutions

NetApp
April 12, 2024

目錄

TR-4896：Azure的分散式訓練：線道偵測-解決方案設計	1
解決方案總覽	1
線道偵測—採用RUN：AI的分散式訓練	3
結論	27
其他資訊	27

TR-4896：Azure的分散式訓練：線道偵測-解決方案設計

NetApp Ronnen Dar、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。

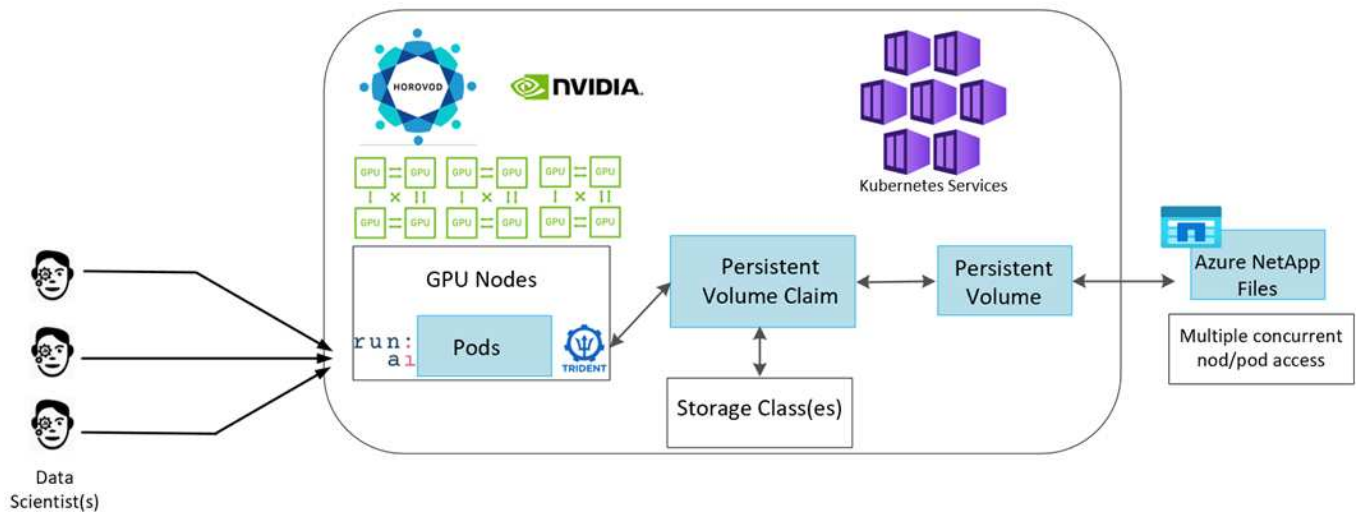
解決方案技術

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
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：

- 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。
- 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 ↑↓
<input type="checkbox"/> 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

在高效能上部署及設定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azurenetaFiles、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 azurenetaFiles 4m47s
```

- a. 檢查執行中的工作：AI UI（或「app.run.ai」）。

run: a Jobs										Cluster: runalaks	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				
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
```

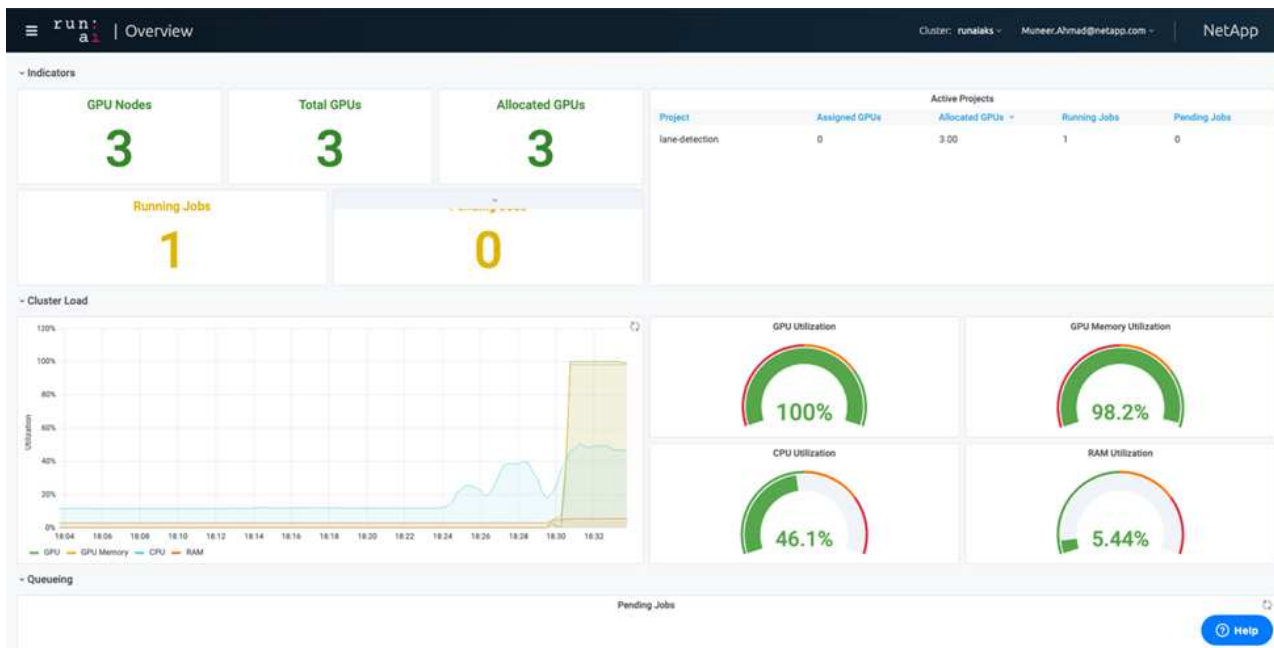
NAME	SERVICE URL(S)	STATUS	AGE	NODE	IMAGE	TYPE	PROJECT	USER	GPUs Allocated (Requested)	PODs
download-tusimple-data		Succeeded	1d		muneer7589/download-tusimple:1.0	Train	lane-detection	verronmartina	0 (0)	0 (0)
dist-lane-detection-training		Init:0/1	2m	<multiple>	muneer7589/dist-lane-detection:3.1	Train	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. The main table lists jobs with columns: Job Name, Status, User, Project, and Total Run Time. A job named 'dist-lane-detection-training' is highlighted with a 'Running' status. To the right, a detailed view of this job is shown, including a 'GPUs' tab with a table of GPU utilization and memory usage.

GPU ↓	Node	Utilization	GPU Memory	Used GPU Memory
0	aks-gpu-pool-34613062...	100%	15.9 GB	15.61 GB
0	aks-gpu-pool-34613062...	100%	15.9 GB	15.61 GB
0	aks-gpu-pool-34613062...	100%	15.9 GB	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" 。

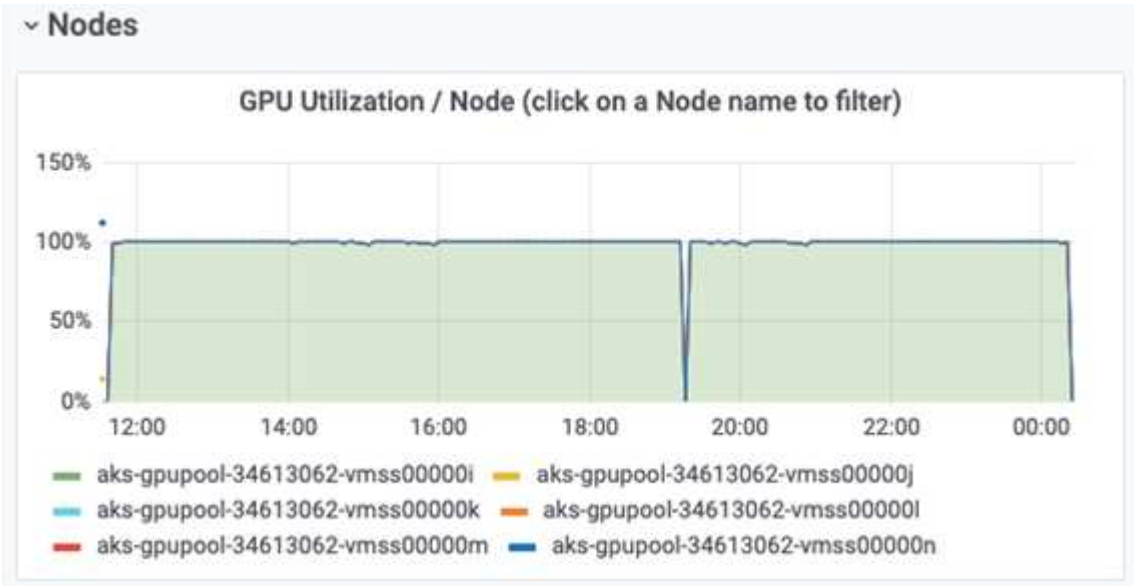
單一GPU的處理時間為12小時45分鐘。三個節點上的三個GPU處理時間約為4小時30分鐘。

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。

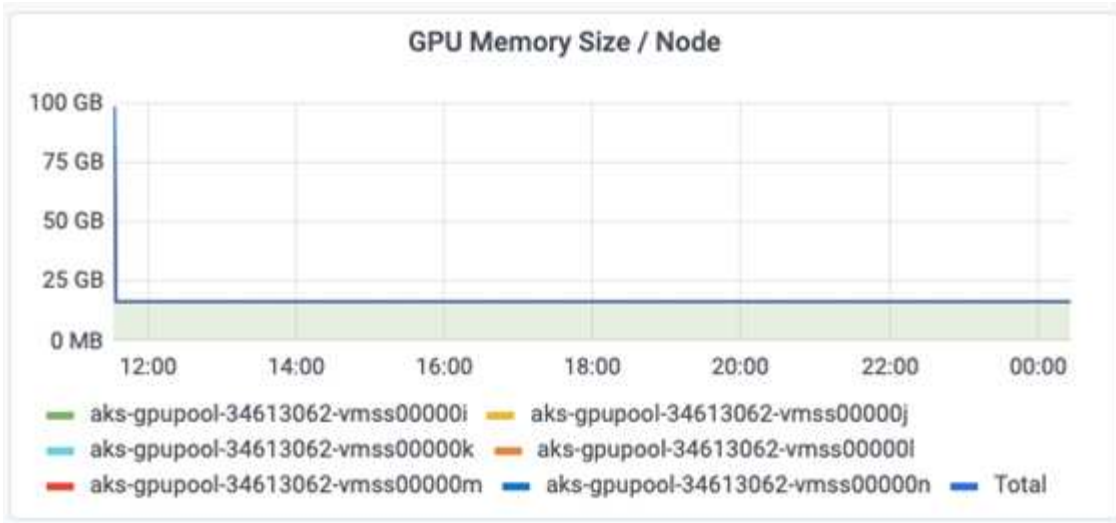
下圖說明1 GPU配置與記憶體使用率。



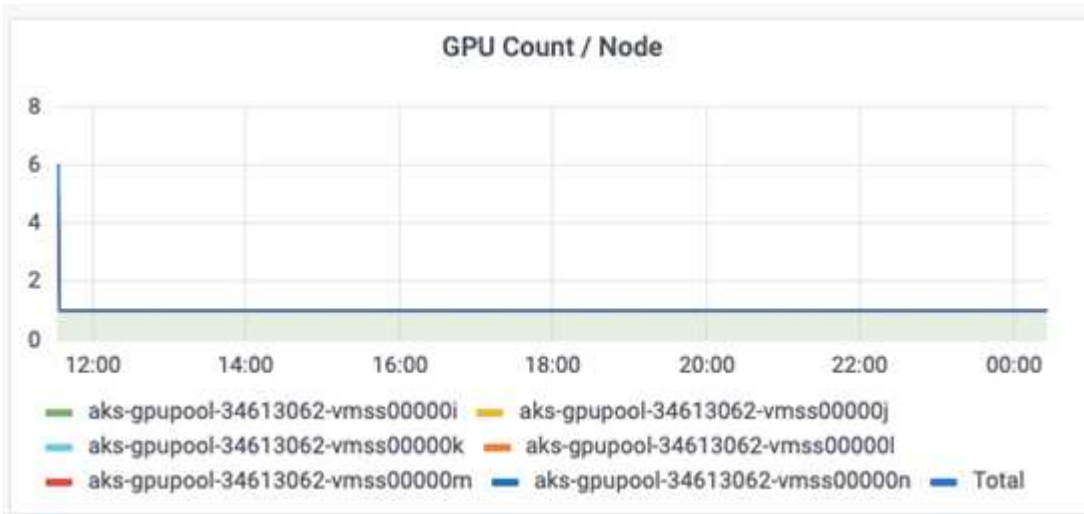
下圖說明單一節點GPU使用率。



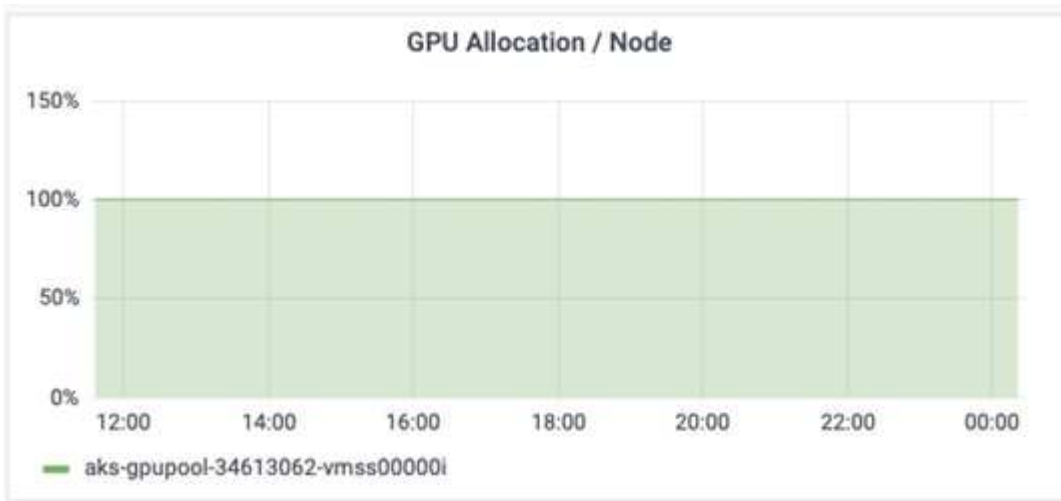
下圖說明單一節點記憶體大小（16GB）。



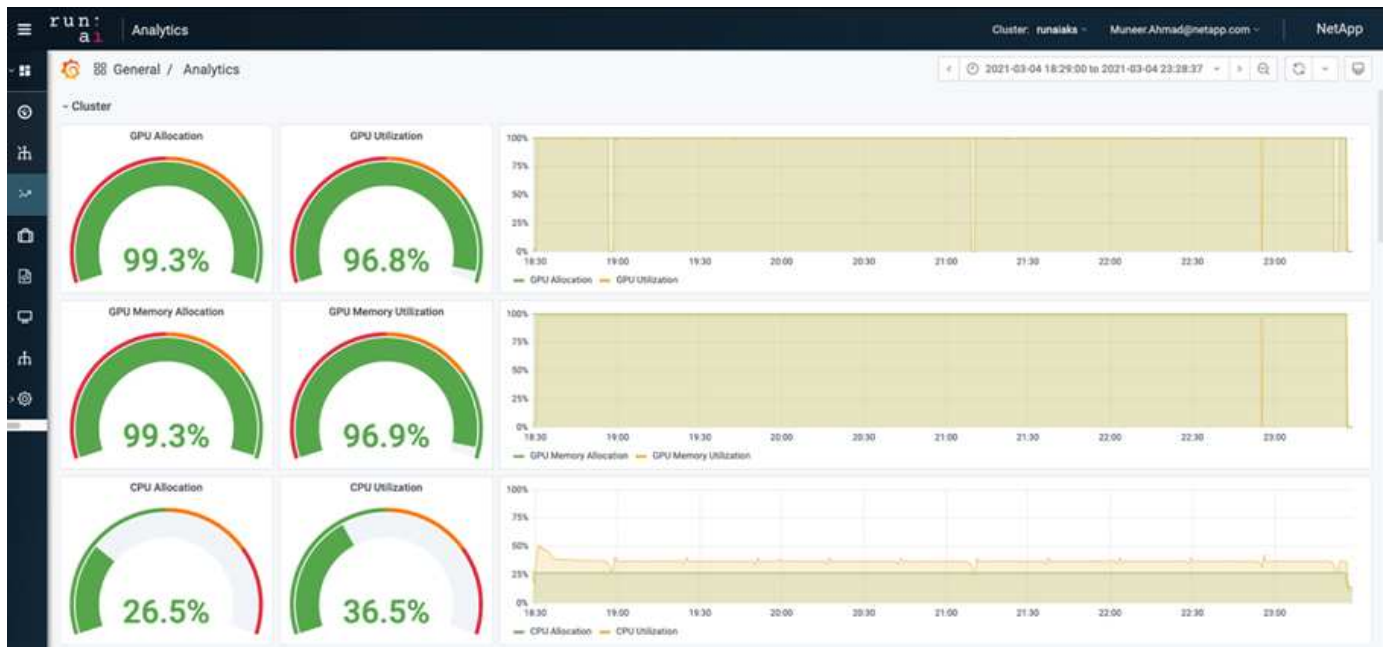
下圖說明單一節點GPU數（1）。



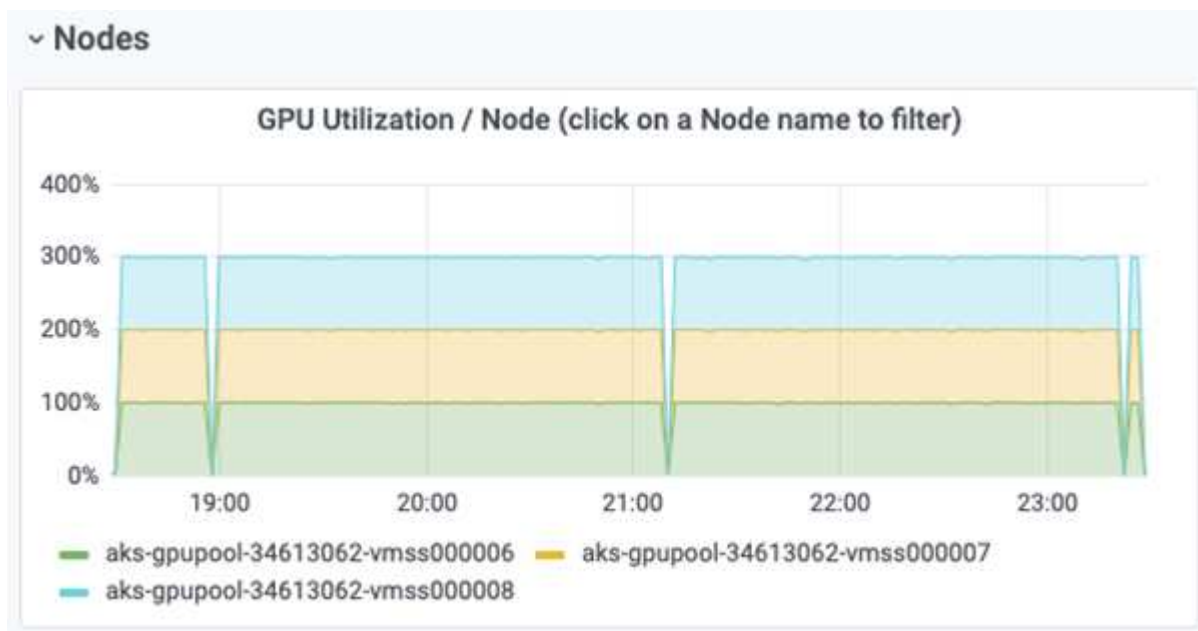
下圖說明單一節點GPU配置（%）。



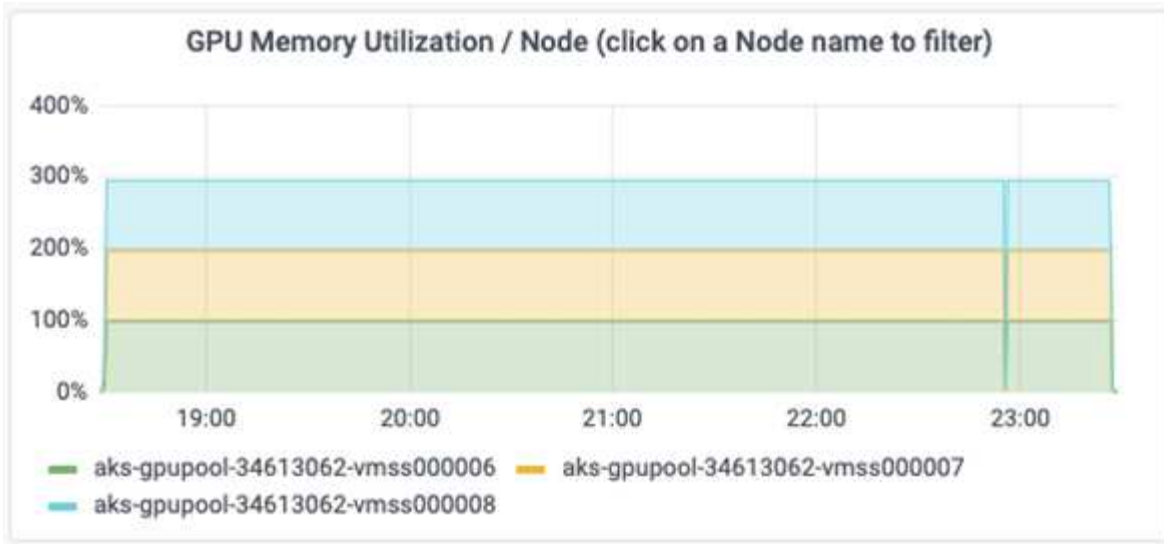
下圖說明三個節點的三個GPU：GPU配置與記憶體。



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

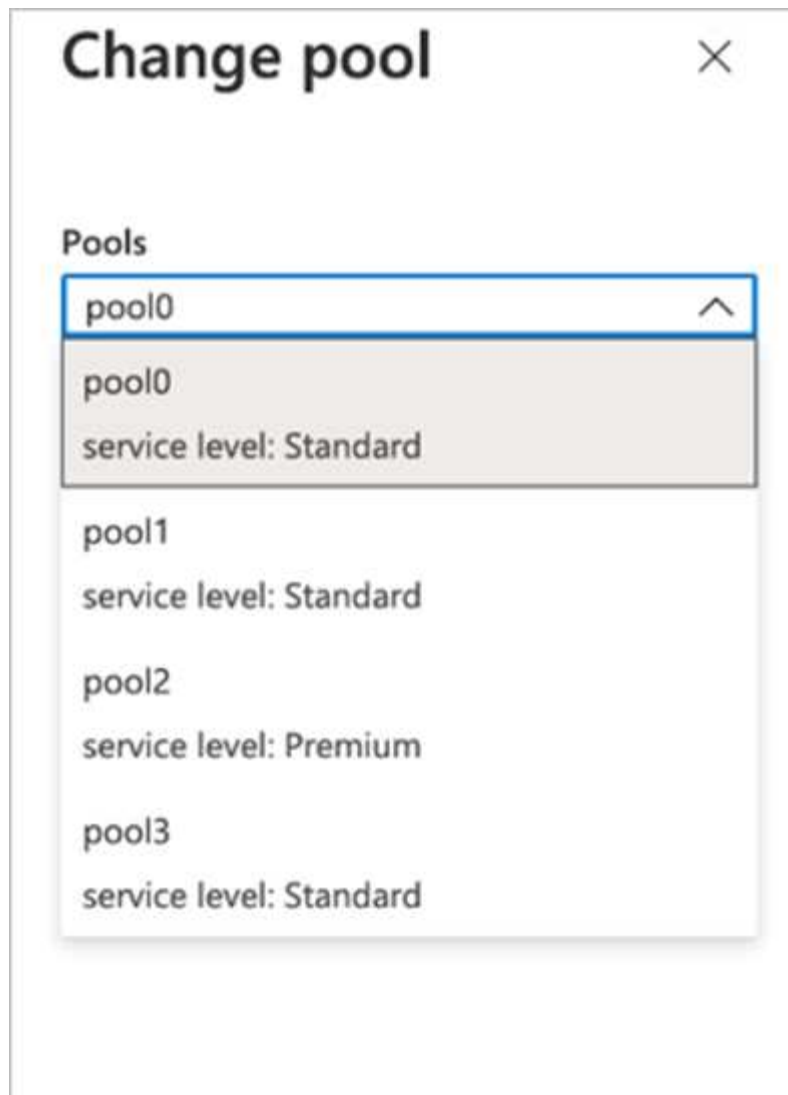
若要變更 Volume 的服務層級、請執行下列步驟：

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

Resize
 Edit
 Change pool
 Delete

2. 在「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/)

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。