



Formazione distribuita su Azure - rilevamento di corsia

NetApp Solutions

NetApp
April 26, 2024

Sommario

- TR-4896: Formazione distribuita in Azure: Rilevamento della corsia - progettazione della soluzione. 1
 - Panoramica della soluzione 1
 - Lane Detection – formazione distribuita con RUN:ai 3
 - Conclusione 29
 - Ulteriori informazioni 29

TR-4896: Formazione distribuita in Azure: Rilevamento della corsia - progettazione della soluzione

Muneer Ahmad e Verron Martina, NetApp Ronen Dar, RUN:ai

Da maggio 2019, Microsoft offre un servizio di portale nativo di Azure per i file service NFS e SMB aziendali basati sulla tecnologia NetApp ONTAP. Questo sviluppo è guidato da una partnership strategica tra Microsoft e NetApp e estende ulteriormente la portata dei servizi dati ONTAP di livello mondiale ad Azure.

NetApp, un provider leader di servizi dati cloud, ha collaborato con RUN: Ai, un'azienda che virtualizza l'infrastruttura ai, per consentire una sperimentazione ai più rapida con un utilizzo completo della GPU. La partnership consente ai team di accelerare l'ai eseguendo numerosi esperimenti in parallelo, con accesso rapido ai dati e sfruttando risorse di calcolo illimitate. RUN: L'ai consente l'utilizzo completo della GPU automatizzando l'allocazione delle risorse, mentre l'architettura comprovata di Azure NetApp Files consente di eseguire ogni esperimento alla massima velocità eliminando le ostruzioni della pipeline dei dati.

NetApp e RUN: L'ai ha Unito le forze per offrire ai clienti una piattaforma a prova di futuro per il loro viaggio nell'ai in Azure. Dagli analytics al calcolo ad alte performance (HPC) alle decisioni autonome (in cui i clienti possono ottimizzare i propri investimenti IT pagando solo ciò di cui hanno bisogno, quando ne hanno bisogno), l'alleanza tra NetApp e RUN: L'ai offre una singola esperienza unificata in Azure Cloud.

Panoramica della soluzione

In questa architettura, l'attenzione si concentra sulla parte più intensiva dal punto di vista computazionale del processo di training distribuito ai o di machine learning (ML) del rilevamento di corsia. Il rilevamento della corsia è una delle attività più importanti nella guida autonoma, che aiuta a guidare i veicoli attraverso la localizzazione delle linee di demarcazione della corsia. Componenti statici come le linee di demarcazione della corsia guidano il veicolo a guidare in autostrada in modo interattivo e sicuro.

Gli approcci convoluzionali basati sulla rete neurale (CNN) hanno portato la comprensione e la segmentazione della scena a un nuovo livello. Anche se non funziona bene per oggetti con strutture e zone lunghe che potrebbero essere occluse (ad esempio, poli, ombre sulla corsia e così via). La rete neurale convoluzionale spaziale (SCNN) generalizza la CNN a un livello spaziale ricco. Consente la propagazione delle informazioni tra neuroni nello stesso livello, il che lo rende più adatto per oggetti strutturati come corsie, pali o camion con occlusioni. Questa compatibilità è dovuta al fatto che le informazioni spaziali possono essere rafforzate e preservano uniformità e continuità.

Migliaia di immagini di scena devono essere iniettate nel sistema per consentire al modello di apprendere e distinguere i vari componenti del set di dati. Queste immagini includono condizioni meteo, diurne o notturne, strade a più corsie e altre condizioni di traffico.

Per la formazione, è necessario disporre di una buona qualità e quantità di dati. Una singola GPU o più GPU possono richiedere da giorni a settimane per completare il training. La formazione distribuita sui dati può accelerare il processo utilizzando GPU multiple e multinode. Horovod è un framework di questo tipo che garantisce la formazione distribuita, ma la lettura dei dati tra cluster di GPU potrebbe costituire un ostacolo. Azure NetApp Files offre un throughput ultraveloce e elevato e una latenza ridotta e sostenuta per fornire

funzionalità scale-out/scale-up in modo che le GPU vengano sfruttate al meglio della loro capacità di calcolo. I nostri esperimenti hanno verificato che tutte le GPU nel cluster vengono utilizzate in media più del 96% per l'addestramento del rilevamento di corsia mediante SCNN.

Pubblico di riferimento

La scienza dei dati incorpora diverse discipline nell'IT e nel business, pertanto più persone fanno parte del nostro pubblico di riferimento:

- Gli scienziati dei dati hanno bisogno della flessibilità necessaria per utilizzare gli strumenti e le librerie di loro scelta.
- I data engineer devono sapere come i dati scorrono e dove risiedono.
- Esperti di casi d'utilizzo per la guida autonoma.
- Amministratori e architetti del cloud per configurare e gestire le risorse cloud (Azure).
- Un tecnico DevOps ha bisogno dei tool per integrare le nuove applicazioni ai/ML nelle pipeline di integrazione continua e implementazione continua (ci/CD).
- Gli utenti aziendali desiderano avere accesso alle applicazioni ai/ML.

In questo documento, descriviamo in che modo Azure NetApp Files, RUN: Ai e Microsoft Azure aiutano ciascuno di questi ruoli a portare valore al business.

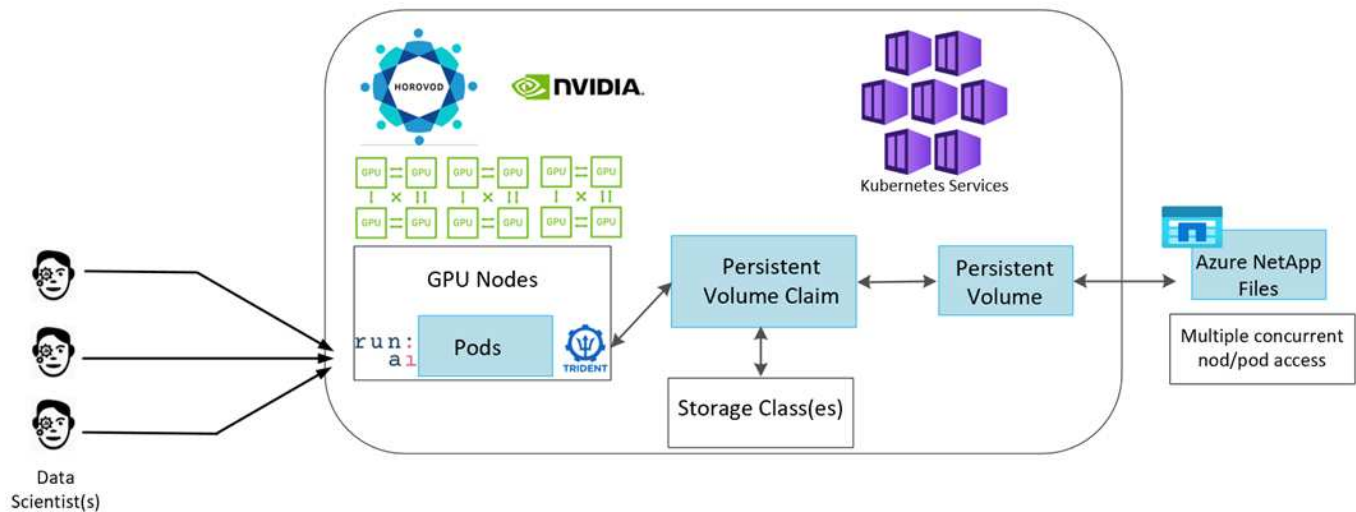
Tecnologia della soluzione

In questa sezione vengono illustrati i requisiti tecnologici per il caso di utilizzo del rilevamento di corsia implementando una soluzione di training distribuita su larga scala che viene eseguita completamente nel cloud Azure. La figura seguente fornisce una panoramica dell'architettura della soluzione.

Gli elementi utilizzati in questa soluzione sono:

- Servizio Azure Kubernetes (AKS)
- Azure Compute SKU con GPU NVIDIA
- Azure NetApp Files
- ESECUZIONE: AI
- Trident di NetApp

I collegamenti a tutti gli elementi menzionati sono elencati nella ["Ulteriori informazioni"](#) sezione.



Requisiti di risorse e servizi cloud

La seguente tabella elenca i componenti hardware necessari per implementare la soluzione. I componenti cloud utilizzati in qualsiasi implementazione della soluzione possono variare in base ai requisiti del cliente.

Cloud	Quantità
AKS	Almeno tre nodi di sistema e tre nodi di lavoro GPU
Nodi di sistema delle SKU delle macchine virtuali (VM)	Tre Standard_DS2_v2
Nodi di lavoro GPU SKU VM	Tre standard_NC6s_v3
Azure NetApp Files	Tier standard da 4 TB

Requisiti software

La seguente tabella elenca i componenti software necessari per implementare la soluzione. I componenti software utilizzati in qualsiasi implementazione della soluzione possono variare in base ai requisiti del cliente.

Software	Versione o altre informazioni
AKS - versione di Kubernetes	1.18.14
ESEGUI:AI CLI	v2.2.25
RUN:ai Orchestration Kubernetes Operator version	1.0.109
Horovod	0.21.2
Trident di NetApp	20.01.1
Timone	3.0.0

Lane Detection – formazione distribuita con RUN:ai

Questa sezione fornisce dettagli sulla configurazione della piattaforma per l'esecuzione del training distribuito di rilevamento della corsia su larga scala utilizzando

L'ORCHESTRATOR DI intelligenza artificiale. Discutiamo dell'installazione di tutti gli elementi della soluzione e dell'esecuzione del lavoro di training distribuito sulla piattaforma suddetta. IL controllo della versione ML viene completato utilizzando NetApp SnapshotTM collegato a ESPERIMENTI DI RUN: Ai per ottenere la riproducibilità dei dati e dei modelli. IL controllo delle versioni DI ML svolge un ruolo cruciale nel monitoraggio dei modelli, nella condivisione del lavoro tra i membri del team, nella riproducibilità dei risultati, nel passaggio in produzione delle nuove versioni dei modelli e nella provenienza dei dati. NetApp ML version control (Snapshot) è in grado di acquisire versioni point-in-time dei dati, dei modelli addestrati e dei registri associati a ciascun esperimento. Grazie al supporto API completo, è facile da integrare con LA piattaforma DI ESECUZIONE: Ai; devi solo attivare un evento in base allo stato del training. Inoltre, è necessario acquisire lo stato dell'intero esperimento senza modificare nulla nel codice o nei container eseguiti su Kubernetes (K8s).

Infine, questo report tecnico si conclude con la valutazione delle performance su più nodi abilitati alla GPU in AKS.

Training distribuito per il caso di utilizzo del rilevamento di corsia utilizzando il set di dati TuSimple

In questo report tecnico, viene eseguito un training distribuito sul set di dati TuSimple per il rilevamento della corsia. Horovod viene utilizzato nel codice di training per condurre training distribuiti su più nodi GPU contemporaneamente nel cluster Kubernetes tramite AKS. Il codice viene confezionato come immagini container per il download e l'elaborazione dei dati TuSimple. I dati elaborati vengono memorizzati su volumi persistenti allocati dal plug-in di NetApp Trident. Per il training, viene creata un'altra immagine container che utilizza i dati memorizzati nei volumi persistenti creati durante il download dei dati.

Per inviare i dati e il lavoro di training, utilizza RUN: Ai per orchestrare l'allocazione e la gestione delle risorse. ESEGUI: L'ai consente di eseguire operazioni MPI (message Passing Interface) necessarie per Horovod. Questo layout consente a più nodi GPU di comunicare tra loro per aggiornare i pesi di training dopo ogni mini batch di training. Consente inoltre di monitorare la formazione attraverso l'interfaccia utente e la CLI, semplificando il monitoraggio dei progressi degli esperimenti.

NetApp Snapshot è integrato nel codice di training e acquisisce lo stato dei dati e il modello formativo per ogni esperimento. Questa funzionalità consente di tenere traccia della versione dei dati e del codice utilizzati e del modello di formazione associato generato.

Installazione e configurazione di AKS

Per la configurazione e l'installazione del cluster AKS, visitare il sito Web all'indirizzo ["Creare un cluster AKS"](#). Quindi, attenersi alla seguente serie di passaggi:

1. Quando si seleziona il tipo di nodi (che si tratti di nodi di sistema (CPU) o di lavoro (GPU)), selezionare quanto segue:
 - a. Aggiungere il nodo di sistema primario denominato `agentpool` su `Standard_DS2_v2` dimensione. Utilizzare i tre nodi predefiniti.
 - b. Aggiungere nodo di lavoro `gpupool` con the `Standard_NC6s_v3` dimensioni del pool. Utilizzare almeno tre nodi per i nodi GPU.

+ 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



L'implementazione richiede 10 minuti.

- Al termine dell'implementazione, fare clic su Connect to Cluster (Connetti al cluster). Per connettersi al cluster AKS appena creato, installare il tool della riga di comando Kubernetes dall'ambiente locale (laptop/PC). Visitare il sito ["Strumenti di installazione"](#) Per installarlo in base al sistema operativo in uso.
- ["Installare Azure CLI nell'ambiente locale"](#).
- Per accedere al cluster AKS dal terminale, immettere `az login` e inserire le credenziali.
- Eseguire i due comandi seguenti:

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

- Immettere questo comando nella riga di comando Azure:

```
kubectl get nodes
```



Se tutti e sei i nodi sono attivi e in esecuzione, il cluster AKS è pronto e connesso all'ambiente locale.

```
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 ~ %
```

Creare una subnet delegata per Azure NetApp Files

Per creare una subnet delegata per Azure NetApp Files, seguire questa serie di passaggi:

- Accedere alle reti virtuali all'interno del portale Azure. Trova la tua rete virtuale appena creata. Dovrebbe avere un prefisso come aks-vnet, come mostrato qui. Fare clic sul nome della rete virtuale.

Microsoft Azure

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

Dashboard > Virtual networks

seanilucelive (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. Fare clic su subnet e selezionare +Subnet nella barra degli strumenti superiore.

Microsoft Azure

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

Dashboard > Virtual networks > aks-vnet-22885919

aks-vnet-22885919 | Subnets

Virtual network

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

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

Search subnets

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

3. Specificare un nome per la subnet, ad esempio ANF. sn E sotto l'intestazione Subnet Delegation (delega subnet), selezionare Microsoft.NetApp/volumes. Non cambiare altro. Fare clic su OK.

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

I volumi Azure NetApp Files vengono allocati nel cluster di applicazioni e vengono utilizzati come dichiarazioni di volumi persistenti (PVC) in Kubernetes. A sua volta, questa allocazione ci offre la flessibilità di mappare i volumi a diversi servizi, sia che si trattino di notebook Jupyter, funzioni senza server e così via

Gli utenti dei servizi possono consumare lo storage dalla piattaforma in molti modi. I principali vantaggi di Azure NetApp Files sono:

- Offre agli utenti la possibilità di utilizzare le snapshot.
- Consente agli utenti di memorizzare grandi quantità di dati su volumi Azure NetApp Files.
- Ottenere i vantaggi in termini di performance dei volumi Azure NetApp Files quando si eseguono i modelli su grandi set di file.

Configurazione di Azure NetApp Files

Per completare la configurazione di Azure NetApp Files, è necessario configurarla come descritto in ["QuickStart: Configurazione di Azure NetApp Files e creazione di un volume NFS"](#).

Tuttavia, è possibile omettere la procedura per creare un volume NFS per Azure NetApp Files, poiché si creeranno volumi tramite Trident. Prima di continuare, assicurarsi di disporre di:

1. ["Registrato per Azure NetApp Files e per il provider di risorse NetApp \(tramite la shell cloud di Azure\)"](#).
2. ["Creato un account in Azure NetApp Files"](#).
3. ["Impostare un pool di capacità"](#) (Minimo 4 TiB Standard o Premium a seconda delle esigenze).

Peering della rete virtuale AKS e della rete virtuale Azure NetApp Files

Quindi, eseguire il peer della rete virtuale AKS con Azure NetApp Files VNET seguendo questa procedura:

1. Nella casella di ricerca nella parte superiore del portale Azure, digitare virtual networks (reti virtuali).
2. Fare clic su VNET aks- vnet-name, quindi immettere Peerings nel campo di ricerca.
3. Fare clic su +Add (Aggiungi) e inserire le informazioni fornite nella tabella seguente:

Campo	Valore o descrizione
Nome del collegamento peering	aks-vnet-name_to_an
SubscriptionID	Iscrizione a Azure NetApp Files VNET a cui stai eseguendo il peering
Partner di peering VNET	Azure NetApp Files VNET



Lasciare tutte le sezioni non contrassegnate come predefinite

4. Fare clic su ADD (AGGIUNGI) o su OK per aggiungere il peering alla rete virtuale.

Per ulteriori informazioni, visitare il sito ["Creare, modificare o eliminare un peering di rete virtuale"](#).

Trident

Trident è un progetto open-source che NetApp gestisce per lo storage persistente dei container delle applicazioni. Trident è stato implementato come un provisioning controller esterno che viene eseguito come pod stesso, monitorando i volumi e automatizzando completamente il processo di provisioning.

NetApp Trident consente un'integrazione perfetta con K8s creando e allegando volumi persistenti per l'archiviazione di set di dati di training e modelli di training. Questa funzionalità semplifica l'utilizzo di K8 da parte di data scientist e data engineer senza il fastidio di memorizzare e gestire manualmente i set di dati. Trident elimina inoltre la necessità per i data scientist di imparare a gestire nuove piattaforme dati, poiché integra le attività correlate alla gestione dei dati attraverso l'integrazione API logica.

Installare Trident

Per installare il software Trident, attenersi alla seguente procedura:

1. ["Installare prima il timone"](#).

2. Scaricare ed estrarre il programma di installazione di Trident 21.01.1.

```
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. Modificare la directory in trident-installer.

```
cd trident-installer
```

4. Copia tridentctl a una directory del sistema \$PATH.

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

5. Installare Trident sul cluster K8s con Helm:

a. Cambiare la directory in Helm directory.

```
cd helm
```

b. Installare Trident.

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

c. Verificare lo stato dei pod Trident nel modo consueto di K8s:

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

d. Se tutti i pod sono in funzione, Trident è installato e si è bene andare avanti.

Configurare il back-end Azure NetApp Files e la classe di storage

Per configurare il back-end Azure NetApp Files e la classe di storage, attenersi alla seguente procedura:

1. Tornare alla home directory.

```
cd ~
```

2. Clonare il ["repository di progetto"](#) lane-detection-SCNN-horovod.

3. Accedere alla `trident-config` directory.

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

4. Creare un principio di servizio Azure (il principio di servizio è il modo in cui Trident comunica con Azure per accedere alle risorse Azure NetApp Files).

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

L'output dovrebbe essere simile al seguente esempio:

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

5. Creare il Trident backend `json` file.

6. Utilizzando l'editor di testo preferito, completare i seguenti campi della tabella riportata di seguito all'interno di `anf-backend.json` file.

Campo	Valore
SubscriptionID	Il tuo ID di abbonamento Azure
ID tenant	Il tuo ID tenant Azure (dall'output di <code>az ad sp</code> nel passaggio precedente)
ID cliente	Il tuo appId (dall'output di <code>az ad sp</code> nel passaggio precedente)
ClientSecret	La tua password (dall'output di <code>az ad sp</code> nel passaggio precedente)

Il file dovrebbe essere simile al seguente esempio:

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

7. Chiedere a Trident di creare il back-end Azure NetApp Files in `trident` namespace, utilizzando `anf-backend.json` come il file di configurazione come segue:

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

8. Creare la classe di storage:

- a. Gli utenti K8 eseguono il provisioning dei volumi utilizzando PVC che specificano una classe di storage in base al nome. Chiedere a K8s di creare una classe di storage `azurenetaappfiles`. Questo farà riferimento al back-end Azure NetApp Files creato nel passaggio precedente utilizzando quanto segue:

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

- b. Verificare che la classe di storage venga creata utilizzando il seguente comando:

```
kubectl get sc azurenetaappfiles
```

L'output dovrebbe essere simile al seguente esempio:

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

Implementare e configurare i componenti di snapshot dei volumi su AKS

Se il cluster non viene fornito con i componenti di snapshot del volume corretti, è possibile installare manualmente questi componenti eseguendo i seguenti passaggi:



AKS 1.18.14 non dispone di Snapshot Controller preinstallato.

1. Installare i CRD Snapshot Beta utilizzando i seguenti comandi:

```
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. Installare Snapshot Controller utilizzando i seguenti documenti di GitHub:

```
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. Impostare K8s volumesnapshotclass`Prima di creare uno snapshot di volume "classe di snapshot del volume" deve essere configurato. Creare una classe di snapshot di volume per Azure NetApp Files e utilizzarla per ottenere il controllo delle versioni ML utilizzando la tecnologia NetApp Snapshot. Creare `volumesnapshotclass netapp-csi-snapclass e impostarlo sul valore predefinito `volumesnapshotclass` come tale:

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

L'output dovrebbe essere simile al seguente esempio:

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

4. Verificare che la classe di copia Snapshot del volume sia stata creata utilizzando il seguente comando:

```
kubectl get volumesnapshotclass
```

L'output dovrebbe essere simile al seguente esempio:

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

ESEGUI:installazione ai

Per installare RUN:ai, attenersi alla seguente procedura:

1. "Installare IL cluster RUN:ai su AKS".
2. Accedere a app.runai.ai, fare clic su Create New Project (Crea nuovo progetto) e assegnargli il nome di rilevamento della corsia. Verrà creato uno spazio dei nomi su un cluster K8s a partire da runai- seguito dal nome del progetto. In questo caso, lo spazio dei nomi creato sarà runai-lane-detection.

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. "INSTALLARE RUN:AI CLI".
4. Sul terminale, impostare il rilevamento di corsia come UN progetto di default RUN: Ai utilizzando il seguente comando:

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

L'output dovrebbe essere simile al seguente esempio:

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

5. Creare ClusterRole e ClusterRoleBinding per lo spazio dei nomi del progetto (ad esempio, lane-detection) quindi, l'account di servizio predefinito appartenente a runai-lane-detection lo spazio dei nomi dispone dell'autorizzazione per eseguire le operazioni volumesnapshot operazioni durante l'esecuzione del processo:
 - a. Elencare gli spazi dei nomi per controllarli runai-lane-detection esiste utilizzando questo comando:

```
kubectl get namespaces
```

L'output dovrebbe apparire come nell'esempio seguente:

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. Creare ClusterRole netappsnapshot E ClusterRoleBinding netappsnapshot utilizzando i seguenti comandi:

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

Scaricare ed elaborare il set di dati TuSimple come lavoro RUN:ai

Il processo per scaricare ed elaborare il set di dati TuSimple come UN processo DI ESECUZIONE: Ai è facoltativo. La procedura prevede i seguenti passaggi:

1. Creare e inviare l'immagine del docker o omettere questo passaggio se si desidera utilizzare un'immagine del docker esistente (ad esempio, muneer7589/download-tusimple:1.0)
 - a. Passare alla home directory:


```
cd ~
```

- b. Accedere alla directory dei dati del progetto lane-detection-SCNN-horovod:

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

- c. Modificare `build_image.sh` shell script e modifica il repository di docker in base al tuo. Ad esempio, sostituire `muneer7589` con il nome del repository di docker. È anche possibile modificare il nome e IL TAG dell'immagine del docker (ad esempio `download-tusimple` e. `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. Eseguire lo script per creare l'immagine del docker e inserirla nel repository del docker utilizzando i seguenti comandi:

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

2. Inviare il lavoro DI ESECUZIONE: Ai per scaricare, estrarre, pre-elaborare e memorizzare il set di dati di rilevamento della corsia TuSimple in un `pvc`, Creata dinamicamente da NetApp Trident:

- a. Utilizzare i seguenti comandi per inviare LA SERIOGRAFIA: AI job:

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

b. Inserire le informazioni dalla tabella seguente per inviare il job RUN:ai:

Campo	Valore o descrizione
-name	Nome del lavoro
pvc	PVC del formato [StorageClassName]:Size:ContainerMountPath nell'invio del job di cui sopra, si sta creando un PVC basato su richiesta utilizzando Trident con azurenetappfile di classe storage. La capacità del volume persistente qui è di 100 Gi ed è montata in path /mnt.
-immagine	Immagine Docker da utilizzare durante la creazione del contenitore per questo lavoro

L'output dovrebbe essere simile al seguente esempio:

```
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. Elencare i job RUN:ai inviati.

```
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)
1 (0)
```

d. Controllare i log dei lavori inoltrati.

```
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
```

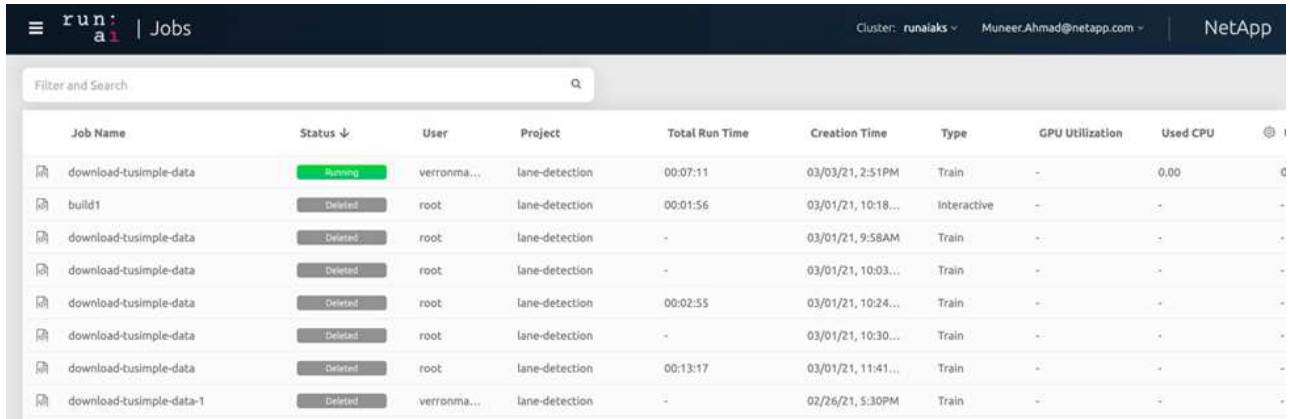
e. Elencare pvc creato. Utilizzare questo pvc comando per la formazione nella fase successiva.

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

L'output dovrebbe essere simile al seguente esempio:

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

- a. Controllare il lavoro IN ESECUZIONE: Ai UI (o. `app.run.ai`).



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	-	-	-

Eseguire un training di rilevamento di corsia distribuito utilizzando Horovod

L'esecuzione di un training di rilevamento di corsia distribuito con Horovod è un processo facoltativo. Tuttavia, di seguito sono riportati i passaggi:

1. Creare e inviare l'immagine del docker o saltare questo passaggio se si desidera utilizzare l'immagine del docker esistente (ad esempio, `muneer7589/dist-lane-detection:3.1`) :

- a. Passare alla home directory.

```
cd ~
```

- b. Accedere alla directory del progetto `lane-detection-SCNN-horovod`.

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

- c. Modificare il `build_image.sh` shell script e modifica il repository di docker in base al tuo (ad esempio, sostituire `muneer7589` con il nome del repository del docker). È anche possibile modificare il nome e IL TAG dell'immagine del docker (`dist-lane-detection e. 3.1, for example`).

```
#!/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. Eseguire lo script per creare l'immagine del docker e passare al repository del docker.

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

2. Inviare la CORSA: Lavoro ai per l'esecuzione del training distribuito (MPI):

- a. Utilizzo di submit of RUN: L'ai per la creazione automatica del PVC nella fase precedente (per il download dei dati) consente solo l'accesso RWO, che non consente a più pod o nodi di accedere allo stesso PVC per la formazione distribuita. Aggiornare la modalità di accesso a ReadWriteMany e utilizzare la patch Kubernetes per eseguire questa operazione.
- b. Innanzitutto, ottenere il nome del volume del PVC eseguendo il seguente comando:

```
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
```

- c. Applicare la patch al volume e aggiornare la modalità di accesso a ReadWriteMany (sostituire il nome del volume con il proprio nel seguente comando):

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

- d. Inviare la CORSA: Lavoro ai MPI per l'esecuzione del lavoro di training distribuito` utilizzando le informazioni della tabella seguente:

```

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"

```

Campo	Valore o descrizione
nome	Nome del lavoro di formazione distribuito
grande shm	Montare un grande dispositivo /dev/shm si tratta di un file system condiviso montato sulla RAM e fornisce una memoria condivisa abbastanza grande per consentire a più lavoratori della CPU di elaborare e caricare batch nella RAM della CPU.
processi	Numero di processi di formazione distribuiti
gpu	Numero di GPU/processi da allocare per il processo in questo processo, esistono tre processi di lavoro GPU (--processi=3), ciascuno allocato con una singola GPU (--gpu 1)
pvc	Utilizza il volume persistente esistente (pvc-download-tusimplici-data-0) creato dal job precedente (download-tusimplici-data) e viene montato nel percorso /mnt
immagine	Immagine Docker da utilizzare durante la creazione del contenitore per questo lavoro
Definire le variabili di ambiente da impostare nel container	
LAVORATORI_DI_UTILIZZO	Impostando l'argomento su true si attiva il caricamento dei dati multi-processo
NUM_WORKERS	Numero di processi di lavoro del data loader
BATCH_SIZE	Dimensione del batch di training
VALORE_UTILIZZO	L'impostazione dell'argomento su true consente la convalida
VAL_BATCH_SIZE	Dimensione del batch di convalida

Campo	Valore o descrizione
ENABLE_SNAPSHOT	Impostando l'argomento su true, è possibile acquisire dati e snapshot dei modelli con formazione per scopi di versioning ML
NOME_PVC	Nome del pvc di cui eseguire un'istantanea. Nell'invio del job di cui sopra, si sta prendendo un'istantanea di pvc-download-tusSimple-data-0, che consiste di dataset e modelli addestrati

L'output dovrebbe essere simile al seguente esempio:

```
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. Elencare il lavoro inoltrato.

```
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. Log dei lavori inoltrati:

```
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. Controllare il lavoro di training in CORSO: Ai GUI (o app.runai.ai): RUN: Ai Dashboard, come mostrato nelle figure seguenti. La prima figura descrive in dettaglio tre GPU allocate per il lavoro di training distribuito su tre nodi su AKS e la seconda ESECUZIONE:job ai:



Jobs

Job Name	Status	User	Project	Total Run Time
dist-lane-detection-training	Running	root	lane-detection	00:12:21
dist-lane-detection-training	Succeeded	verronma...	lane-detection	00:05:04
download-tusimple-data	Succeeded	verronma...	lane-detection	06:00:32
build1	Deleted	root	lane-detection	00:01:56
dist-lane-detection-training	Deleted	verronma...	lane-detection	00:13:36
download-tusimple-data	Deleted	root	lane-detection	-
download-tusimple-data	Deleted	root	lane-detection	-
download-tusimple-data	Deleted	root	lane-detection	-
download-tusimple-data	Deleted	root	lane-detection	00:02:55
download-tusimple-data	Deleted	root	lane-detection	00:13:17
download-tusimple-data-1	Deleted	verronma...	lane-detection	-

dist-lane-detection-training

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. Al termine del training, controlla la copia Snapshot di NetApp creata e collegata al lavoro RUN: Ai.

```
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
```

Ripristinare i dati dalla copia Snapshot di NetApp

Per ripristinare i dati dalla copia Snapshot di NetApp, attenersi alla seguente procedura:

1. Passare alla home directory.

```
cd ~
```

2. Accedere alla directory del progetto lane-detection-SCNN-horovod.

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

3. Modificare restore-snapshot-pvc.yaml e aggiornare dataSource name Nella copia Snapshot da cui si desidera ripristinare i dati. È anche possibile modificare il nome PVC in cui verranno ripristinati i dati, in questo esempio ITS restored-tusimple.

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

4. Creare un nuovo PVC utilizzando restore-snapshot-pvc.yaml.

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

L'output dovrebbe essere simile al seguente esempio:

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

5. Se si desidera utilizzare i dati appena ripristinati per la formazione, l'invio del lavoro rimane lo stesso di prima; sostituire solo PVC_NAME con il ripristinato PVC_NAME quando si invia il lavoro di formazione, come indicato nei seguenti comandi:


```
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"
```

Valutazione delle performance

Per mostrare la scalabilità lineare della soluzione, sono stati eseguiti test delle performance per due scenari: Una GPU e tre GPU. L'allocazione della GPU, l'utilizzo della GPU e della memoria, diverse metriche a nodo singolo e a tre nodi sono state acquisite durante il training sul set di dati di rilevamento della corsia TuSimple. I dati vengono aumentati di cinque volte solo per analizzare l'utilizzo delle risorse durante i processi di training.

La soluzione consente ai clienti di iniziare con un piccolo set di dati e poche GPU. Quando la quantità di dati e la domanda di GPU aumentano, i clienti possono scalare dinamicamente i terabyte nel Tier Standard e scalare rapidamente fino al Tier Premium per ottenere un throughput quattro volte superiore per terabyte senza spostare alcun dato. Questo processo viene spiegato ulteriormente nella sezione, "[Livelli di servizio Azure NetApp Files](#)".

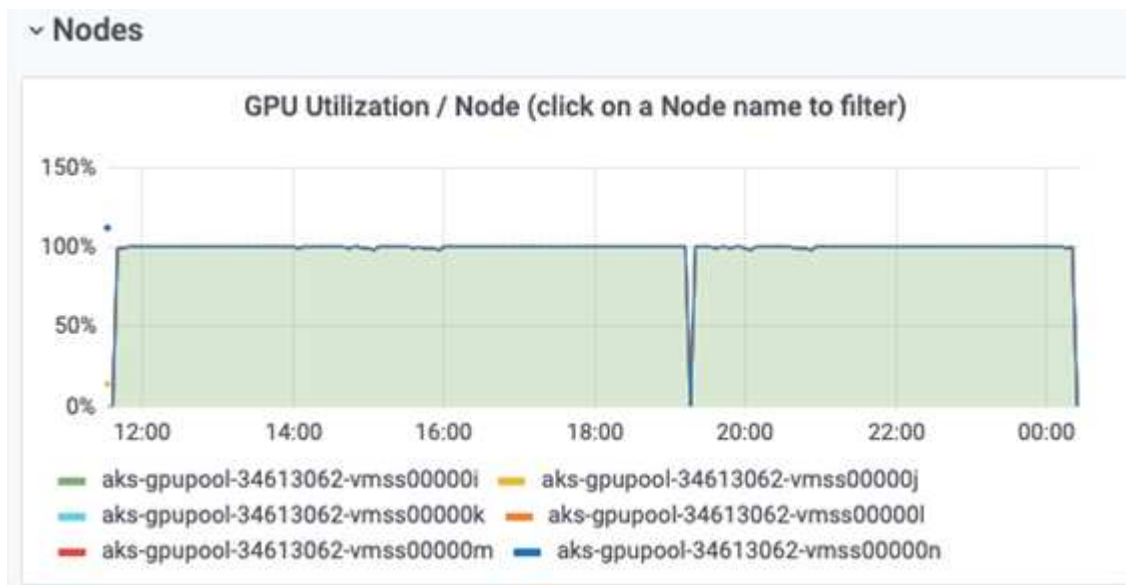
Il tempo di elaborazione su una GPU era di 12 ore e 45 minuti. Il tempo di elaborazione su tre GPU su tre nodi era di circa 4 ore e 30 minuti.

Le figure mostrate nel resto di questo documento illustrano esempi di performance e scalabilità in base alle singole esigenze aziendali.

La figura seguente illustra l'allocazione e l'utilizzo della memoria di 1 GPU.



La figura seguente illustra l'utilizzo della GPU a nodo singolo.



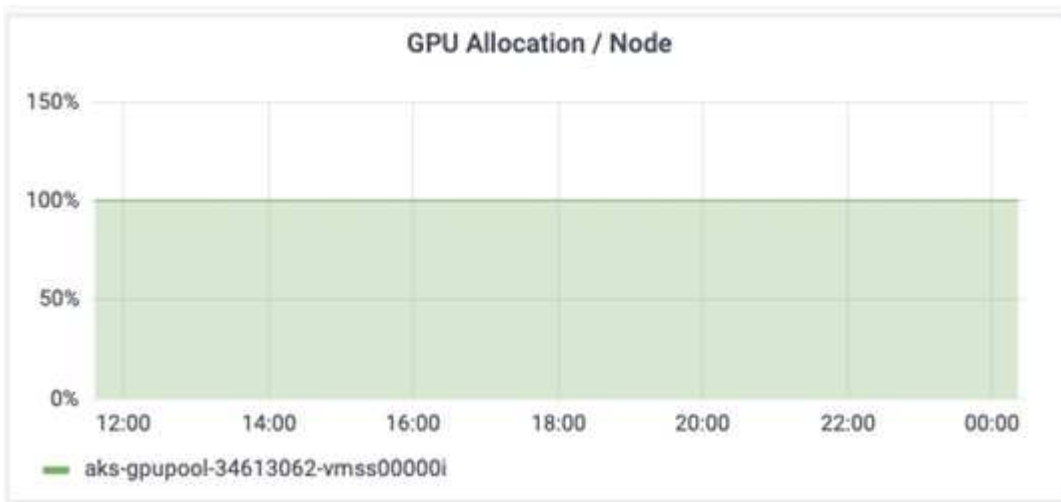
La figura seguente illustra le dimensioni della memoria a nodo singolo (16 GB).



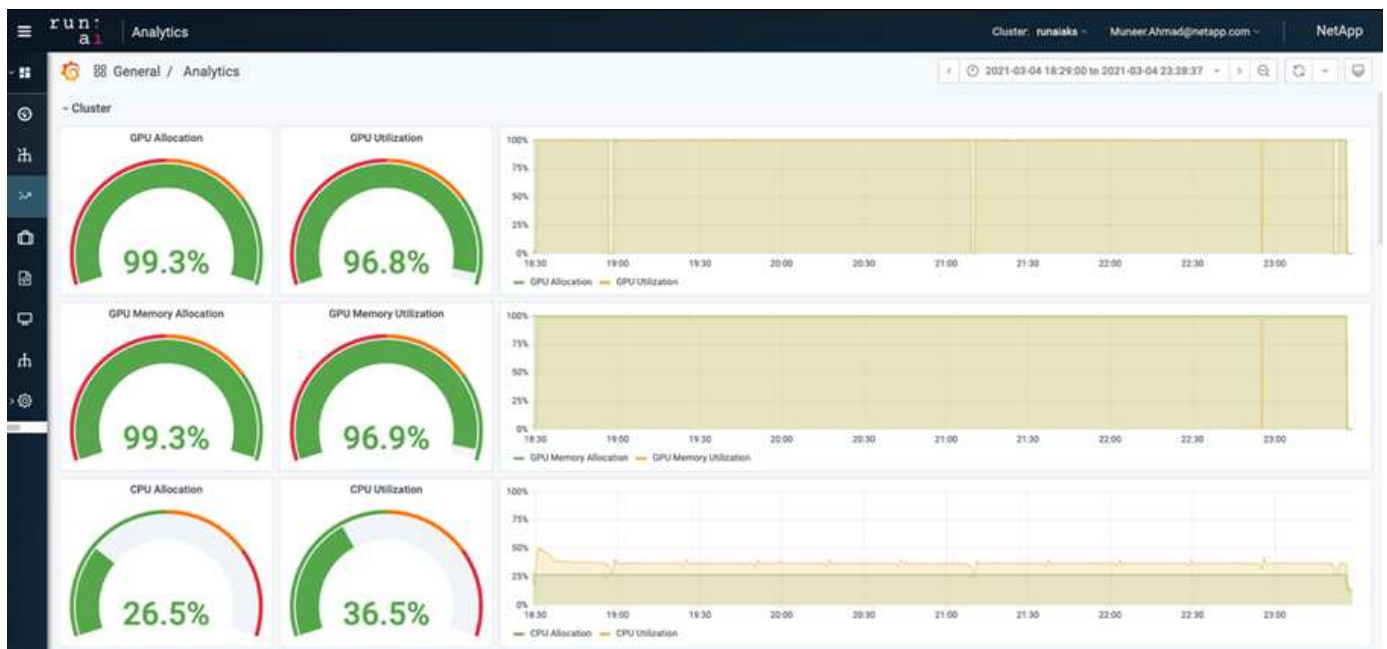
La figura seguente illustra il numero di GPU a nodo singolo (1).



La figura seguente illustra l'allocazione della GPU a nodo singolo (%).



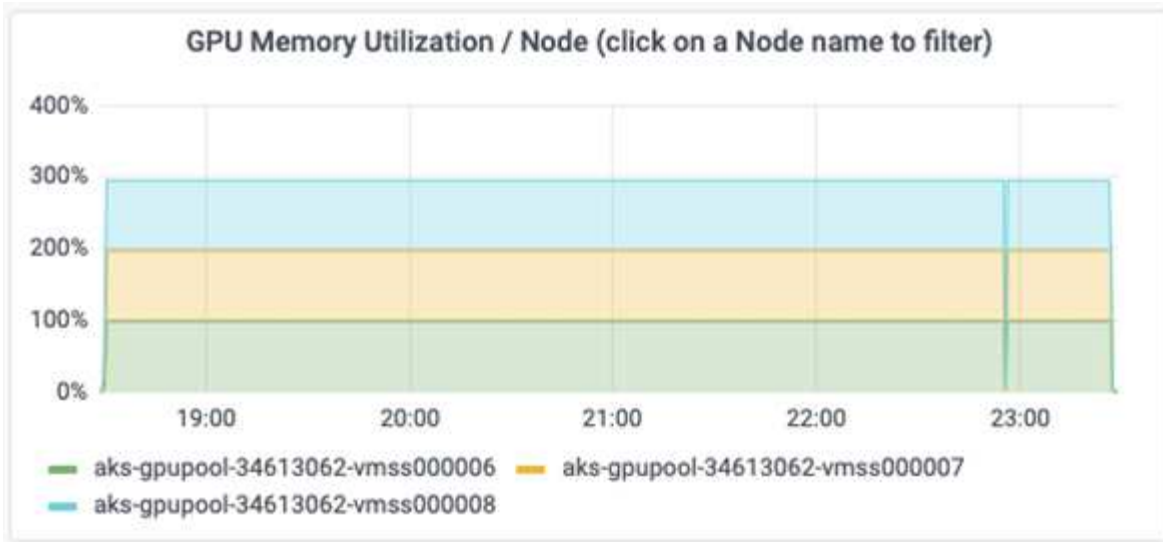
La figura seguente illustra tre GPU su tre nodi: Allocazione e memoria delle GPU.



La figura seguente illustra tre GPU in tre nodi utilizzati (%).



La figura seguente illustra tre GPU in tre nodi di utilizzo della memoria (%).



Livelli di servizio Azure NetApp Files

È possibile modificare il livello di servizio di un volume esistente spostando il volume in un altro pool di capacità che utilizza "livello di servizio" si desidera per il volume. Questa modifica del livello di servizio esistente per il volume non richiede la migrazione dei dati. Inoltre, non influisce sull'accesso al volume.

Modificare dinamicamente il livello di servizio di un volume

Per modificare il livello di servizio di un volume, attenersi alla seguente procedura:

1. Nella pagina Volumes (volumi), fare clic con il pulsante destro del mouse sul volume di cui si desidera modificare il livello di servizio. Selezionare Cambia pool.

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		...

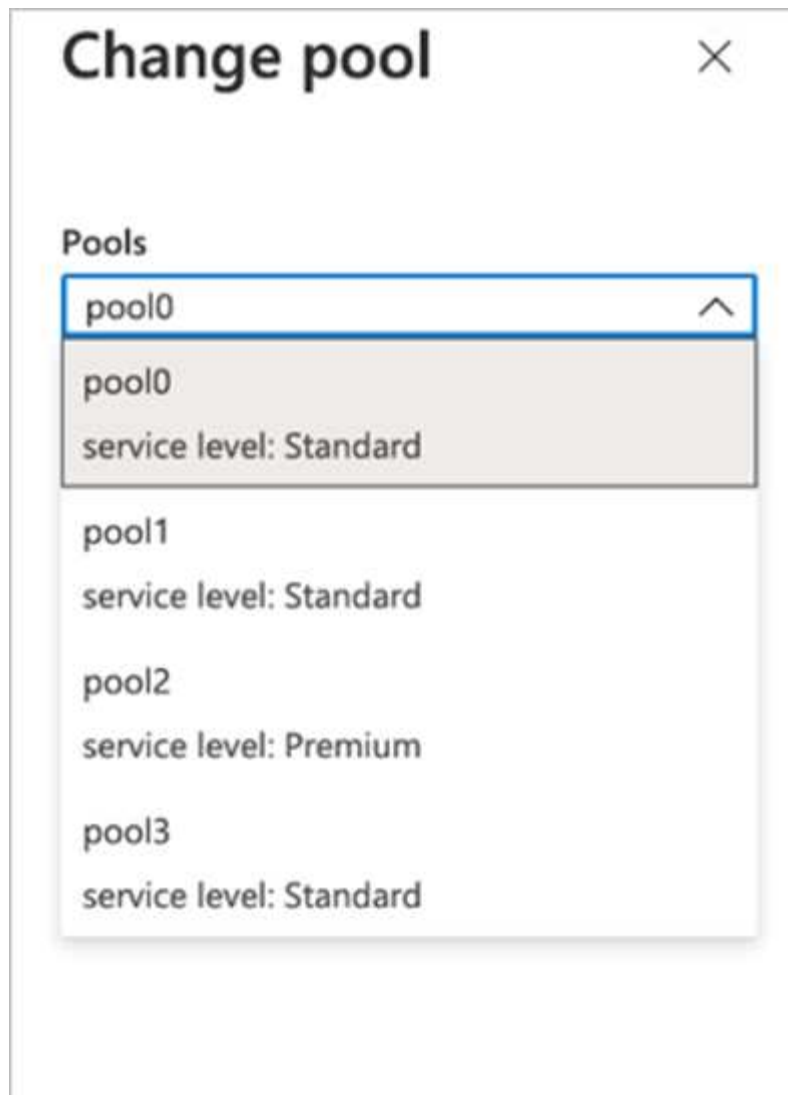
Resize

Edit

Change pool

Delete

2. Nella finestra Change Pool, selezionare il pool di capacità in cui si desidera spostare il volume. Quindi, fare clic su OK.



Automatizzare la modifica del livello di servizio

La modifica dinamica del livello di servizio è ancora in Public Preview, ma non è attivata per impostazione predefinita. Per attivare questa funzione nell'abbonamento Azure, seguire la procedura descritta nel documento "[Modificare dinamicamente il livello di servizio di un volume](#)".

- Per Azure è inoltre possibile utilizzare i seguenti comandi: CLI. Per ulteriori informazioni su come modificare le dimensioni del pool di Azure NetApp Files, visitare il sito "[Volume netappfiles az: Gestione delle risorse dei volumi ANF \(Azure NetApp Files\)](#)".

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

- Il set- `aznetappfilesvolumepool` Il cmdlet illustrato può modificare il pool di un volume Azure NetApp Files. Per ulteriori informazioni sulla modifica delle dimensioni del pool di volumi e di Azure PowerShell, visitare il sito Web "[Modifica del pool per un volume Azure NetApp Files](#)".

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

Conclusione

NetApp e RUN: L'ai ha collaborato alla creazione di questo report tecnico per dimostrare le funzionalità uniche di Azure NetApp Files insieme alla piattaforma RUN: Ai per semplificare l'orchestrazione dei carichi di lavoro ai. Questo report tecnico fornisce un'architettura di riferimento per semplificare il processo di pipeline di dati e orchestrazione dei carichi di lavoro per il training di rilevamento della corsia distribuita.

In conclusione, per quanto riguarda la formazione distribuita su larga scala (soprattutto in un ambiente di cloud pubblico), il componente di orchestrazione delle risorse e storage è una parte critica della soluzione. Assicurarci che la gestione dei dati non ostacoli mai l'elaborazione di più GPU, per cui si ottiene un utilizzo ottimale dei cicli GPU. Pertanto, rendendo il sistema il più conveniente possibile per scopi di formazione distribuita su larga scala.

Il data fabric fornito da NetApp supera la sfida consentendo a data scientist e data engineer di connettersi tra loro on-premise e nel cloud per avere dati sincroni, senza eseguire alcun intervento manuale. In altre parole, il data fabric rende più uniforme il processo di gestione del workflow ai distribuito in più sedi. Inoltre, facilita la disponibilità dei dati on-demand avvicinando i dati al calcolo ed eseguendo analisi, training e validazione, dove e quando necessario. Questa funzionalità non solo consente l'integrazione dei dati, ma anche la protezione e la sicurezza dell'intera pipeline di dati.

Ulteriori informazioni

Per ulteriori informazioni sulle informazioni descritte in questo documento, consultare i seguenti documenti e/o siti Web:

- Dataset: TuSimple

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

- Deep Learning Network Architecture: Rete neurale spaziale convoluzionale

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

- Framework distribuito per il deep learning: Horovod

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

- ESEGUI: Soluzione di orchestrazione dei container ai: ESEGUI: Introduzione al prodotto ai

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

- ESEGUI: Documentazione di installazione ai

["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)

- Invio di job in ESECUZIONE: 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/)

- Risorse cloud di Azure: Azure NetApp Files

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

- Servizio Azure Kubernetes

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

- SKU di Azure VM

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

- Macchine virtuali Azure con SKU GPU

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

- Trident di NetApp

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

- Data fabric basato su NetApp

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

- Documentazione sui prodotti NetApp

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

Informazioni sul copyright

Copyright © 2024 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.