Distributed Training in Azure – Click-Through-Rate Prediction

NetApp Solutions

NetApp
April 19, 2024

This PDF was generated from https://docs.netapp.com/de-de/netapp-solutions/ai/aks-anf_introduction.html on April 19, 2024. Always check docs.netapp.com for the latest.

Inhalt

Distributed Training in Azure – Click-Through-Rate Prediction . . . 1
TR-4904: Distributed Training in Azure - Click-Through Rate Prediction . . . 1
Technologieüberblick . . . 2
Softwareanforderungen . . . 4
Anforderungen an die Cloud-Ressourcen . . . 4
Anwendungsbeispiel für die Click-Through-Rate-Prognose . . . 4
Einrichtung . . . 6
Klicken Sie sich durch die Verarbeitung der Tarifprognosen und das Modelltraining . . . 16
Schlussfolgerung . . . 22
Wo Sie weitere Informationen finden . . . 23

Distributed Training in Azure – Click-Through-Rate Prediction

TR-4904: Distributed Training in Azure - Click-Through Rate Prediction

Rick Huang, Verron Martina, Muneer Ahmad, NetApp

Der Fokus der Arbeit eines Datenwissenschaftlers sollte auf das Training und Tuning von ml- (Machine Learning) und künstlicher Intelligenz (KI)-Modellen liegen. Laut einer Untersuchung von Google verbringen Data Scientists jedoch etwa 80 Prozent ihrer Zeit damit, herauszufinden, wie ihre Modelle mit Enterprise-Applikationen funktionieren und im großen Maßstab laufen lassen können.

Für das Management von End-to-End-KI/ML-Projekten ist ein umfassenderes Verständnis von Unternehmenskomponenten erforderlich. Obwohl DevOps die Definition, Integration und Implementierung übernommen hat, zielen diese Komponententypen IM ML-Betrieb auf einen ähnlichen Flow ab, der auch KI-/ML-Projekte umfasst. Eine Vorstellung der Merkmale einer End-to-End-KI/ML-Pipeline im Unternehmen erhalten Sie in der folgenden Liste der erforderlichen Komponenten:

- Storage
- Netzwerkbetrieb
- Datenbanken
- File-Systeme
- Container
- CI/CD-Pipeline (Continuous Integration und Continuous Deployment
- Integrierte Entwicklungsumgebung (IDE)
- Sicherheit
- Datenzugriffsrichtlinien
- Trennt
- Cloud
- Einheitliche
- Toolsets und Bibliotheken für Data Science

Zielgruppe

Die Welt der Datenwissenschaft berührt mehrere Disziplinen in IT und Business:

- Der Data Scientist benötigt die Flexibilität, ihre Tools und Bibliotheken einzusetzen.
- Der Data Engineer muss wissen, wie die Daten fließen und wo sie sich befinden.
- DevOps-Engineers benötigen die Tools, um neue KI/ML-Applikationen in ihre CI/CD-Pipelines zu integrieren.
- Cloud-Administratoren und -Architekten müssen in der Lage sein, Azure-Ressourcen einzurichten und zu

managen.

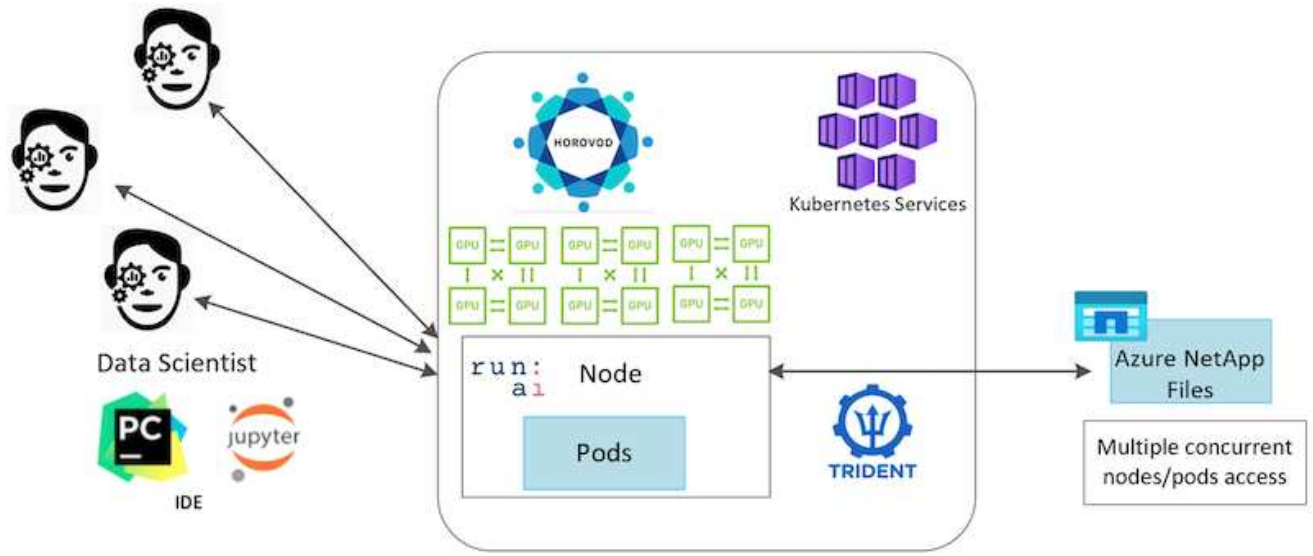
- Geschäftsanwender möchten auf KI-/ML-Applikationen zugreifen können.

In diesem technischen Bericht beschreiben wir, wie Azure NetApp Files, RAPIDS KI, DASK und Azure jede dieser Rollen unterstützen, ihren geschäftlichen Nutzen zu steigern.

Lösungsüberblick

Diese Lösung folgt dem Lebenszyklus einer KI/ML-Applikation. Wir beginnen mit der Arbeit von Data Scientists, um die verschiedenen Schritte zu definieren, die für die Vorbereitung von Daten und das Training von Modellen erforderlich sind. Durch die Nutzung VON RAPIDS on DASK führen wir Distributed Training für den Azure Kubernetes Service (AKS) Cluster durch, um die Trainingszeit im Vergleich zum herkömmlichen Python-Scikit-Learn-Ansatz deutlich zu reduzieren. Um den gesamten Zyklus abzuschließen, haben wir die Pipeline mit Azure NetApp Files integriert.

Azure NetApp Files bietet verschiedene Performance-Tiers. Kunden können mit einer Standardebene beginnen und unterbrechungsfrei auf eine hochperformante Tier skalieren, ohne Daten verschieben zu müssen. Auf diese Weise können Data Scientists Modelle nach Maß ohne Performance-Probleme trainieren und dabei Datensilos im gesamten Cluster vermeiden, wie in der Abbildung unten gezeigt.



Technologieüberblick

Diese Seite bietet einen Überblick über die in dieser Lösung verwendete Technologie.

Microsoft und NetApp

Seit Mai 2019 bietet Microsoft einen nativen Portal-Service auf Basis von Azure, dem First-Party-Anbieter für NFS- und SMB-Fileservices der Enterprise-Klasse, die auf der NetApp ONTAP Technologie basieren. Ziel dieser Entwicklung ist eine strategische Partnerschaft zwischen Microsoft und NetApp, mit der auch die Ausweitung der Reichweite erstklassiger ONTAP-Datenservices auf Azure möglich ist.

Azure NetApp Dateien

Der Azure NetApp Files Service ist ein hochperformanter File Storage-Service der Enterprise-Klasse mit

Messung. Azure NetApp Files unterstützt jeden Workload-Typ und ist standardmäßig hochverfügbar. Sie können Service- und Performance-Level auswählen und Snapshot-Kopien über den Service einrichten. Azure NetApp Files ist ein Azure-Erstanbieter-Service zur Migration und Ausführung der anspruchsvollsten Enterprise-Datei-Workloads in der Cloud, einschließlich Datenbanken, SAP und High-Performance-Computing-Applikationen ohne Codeänderungen.

Diese Referenzarchitektur bietet IT-Abteilungen folgende Vorteile:

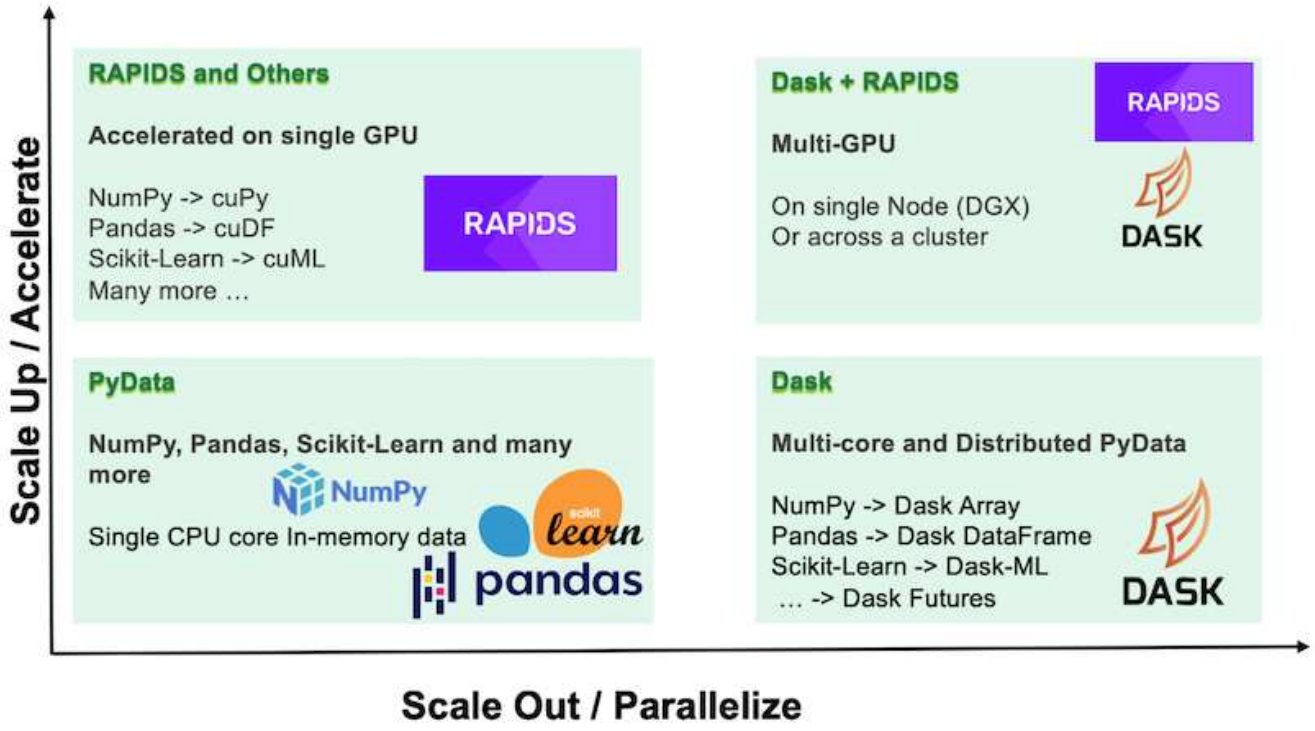
- Sie vereinfacht das Design
- Computing und Storage können unabhängig voneinander skaliert werden
- Kunden können mit einer kleinen Konfiguration starten und nahtlos skalieren
- Bietet eine Reihe von Storage-Tiers für diverse Performance- und Kostenpunkte

Dask und NVIDIA RAPIDS: Überblick

DASK ist ein Open-Source-Tool für paralleles Computing, das Python Bibliotheken auf mehreren Maschinen skaliert und eine schnellere Verarbeitung großer Datenmengen ermöglicht. Es stellt eine API zur Verfügung, die herkömmlichen Python-Bibliotheken mit einem Thread wie Pandas, Numpy und scikit-Learn ähnelt. Daher sind native Python-Benutzer nicht gezwungen, viel am vorhandenen Code zu ändern, um Ressourcen auf dem Cluster zu nutzen.

NVIDIA RAPIDS ist eine Suite aus Open-Source-Bibliotheken, mit denen es möglich ist, vollständig auf GPUs ausgeführte ML- und Data-Analytics-Workflows auszuführen. Zusammen mit DASK können Sie mühelos von der GPU-Workstation (vertikal skalierbar) auf Multi-Node-Cluster mit mehreren GPUs skalieren (horizontal skalierbar).

Für die Implementierung von DASK auf einem Cluster könnte Kubernetes zur Ressourcenorchestrierung verwendet werden. Sie können die Worker-Nodes auch entsprechend den Prozessanforderungen vertikal oder horizontal skalieren und so den Cluster-Ressourcenverbrauch optimieren, wie in der folgenden Abbildung dargestellt.



Softwareanforderungen

In der folgenden Tabelle sind die Softwareanforderungen für diese Lösung aufgeführt.

Software	Version
Azure Kubernetes Service	1.18.14
Container-Image VON RAPIDS und Dask	Projektarchiv: "Rapidsai/rapidsai" Tag: 0.17-cuda11.0-Runtime-ubuntu18.04
NetApp Trident	20.01.1
Helm	3.0.0

Anforderungen an die Cloud-Ressourcen

Auf dieser Seite werden die Konfiguration von Cloud-Ressourcen für Azure NetApp Files beschrieben.

Konfigurieren Sie Azure NetApp Files

Konfigurieren Sie Azure NetApp Files wie unter beschrieben "QuickStart: Azure NetApp Files einrichten und ein NFS-Volume erstellen".

Sie können über den Abschnitt „Create NFS Volume for Azure NetApp Files“ hinaus fortfahren, weil Sie Volumes über Trident erstellen. Bevor Sie fortfahren, gehen Sie wie folgt vor:

1. Registrieren Sie sich über die Azure Shell für Azure NetApp Files und NetApp Ressourcen-Provider ("Verlinken").
2. Erstellen Sie ein Konto in Azure NetApp Files ("Verlinken").
3. Richten Sie einen Kapazitäts-Pool ein (mindestens 4 TB Standard oder Premium, je nach Bedarf) ("Verlinken"In der folgenden Tabelle sind die Anforderungen an die Netzwerkkonfiguration für die Einrichtung in der Cloud aufgeführt. Der DASK-Cluster und der Azure NetApp Files müssen sich im gleichen virtuellen Azure-Netzwerk (vnet) oder einem Peering vnet befinden.

Ressourcen	Typ/Version
Azure Kubernetes Service	1.18.14
Agent-Knoten	3x Standard_DS2_v2
GPU-Node	3x Standard_NC6s_v3
Azure NetApp Dateien	Standard-Kapazitäts-Pool
Kapazität in TB	4

Anwendungsbeispiel für die Click-Through-Rate-Prognose

Dieser Anwendungsfall basiert auf dem öffentlich zugänglichen "Terabyte Auf Protokolle Klicken" Datensatz von "Criteo AI Lab". Aufgrund der jüngsten Fortschritte BEI ML-Plattformen und -Applikationen wird jetzt viel Aufmerksamkeit auf das Lernen in großen

Umgebungen gerichtet. Die Klickrate (Klickrate, CTR) ist definiert als die durchschnittliche Anzahl der Klickrate pro hundert Online-Werbeeindrücke (ausgedrückt als Prozentsatz). Es wird in verschiedenen Branchen und Anwendungsfällen, darunter digitales Marketing, Einzelhandel, E-Commerce und Service-Provider, häufig als wichtige Metrik eingeführt. Beispiele für die Verwendung von CTR als wichtige Kennzahl für potenziellen Kundenverkehr sind:

- **Digital Marketing:** in "Google Analytics", CTR kann verwendet werden, um zu messen, wie gut ein Werbetreibender oder Kaufmann die Keywords, Anzeigen und kostenlose Angebote durchführen. Ein hoher CTR ist ein guter Hinweis darauf, dass Benutzer Ihre Anzeigen und Inserate hilfreich und relevant finden. CTR trägt auch zu Ihrem Keyword erwartet CTR, die eine Komponente von ist "Anzeigeneinstufung".
- **E-Commerce:** Neben der Nutzung "Google Analytics", Es gibt mindestens einige Besucherstatistiken in einem E-Commerce-Backend. Obwohl diese Statistiken auf den ersten Blick nicht nützlich erscheinen, sind sie in der Regel leicht zu lesen und können genauer als andere Informationen sein. Datensätze von Erstanbietern, die sich aus diesen Statistiken zusammensetzt, sind vertraulich und daher am wichtigsten für E-Commerce-Verkäufer, Käufer und Plattformen. Diese Datensätze können zur Festlegung von Benchmarks verwendet und mit dem Ergebnis vom letzten Jahr und von gestern verglichen werden, indem eine Zeitreihe zur weiteren Analyse erstellt wird.
- **Einzelhandel:** stationäre Einzelhändler können die Anzahl der Besucher und die Anzahl der Kunden mit dem CTR korrelieren. Die Anzahl der Kunden kann aus ihrer Verkaufsgeschichte ersichtlich werden. Die CTR von den Webseiten des Einzelhändlers oder Werbeverkehr kann zu den oben genannten Verkäufen führen. Treueprogramme sind ein weiterer Anwendungsfall, weil Kunden von Online-Anzeigen oder anderen Websites umgeleitet werden könnten, um Belohnungen zu verdienen. Einzelhändler können Kunden über Treueprogramme gewinnen und Verhaltensweisen aus Verkaufsprogrammen aufzeichnen, um ein Empfehlungssystem zu entwickeln, das nicht nur das Kaufverhalten der Verbraucher in verschiedenen Kategorien prognostiziert, sondern auch Coupons personalisiert und Abgänge verringert.
- **Service-Provider:** Telekommunikationsunternehmen und Internet-Service-Provider haben eine Fülle an Telemetriedaten von Erstbenutzern für aufschlussreiche KI-, ML- und Analytics-Anwendungsfälle. So kann ein Telekommunikationsanbieter beispielsweise täglich die Protokolle der Top-Level-Domain-Historie seiner mobilen Abonnenten nutzen, um vorhandene Modelle zu optimieren, um eine aktuelle Segmentierung der Zielgruppe zu erstellen, das Kundenverhalten vorherzusagen und mit Werbetreibenden zusammenzuarbeiten, um Anzeigen in Echtzeit zu platzieren, um ein besseres Online-Erlebnis zu ermöglichen. In einem solchen datengesteuerten Marketing-Workflow ist CTR eine wichtige Kennzahl, um Konvertierungen widerzuspiegeln.

Im Rahmen des digitalen Marketings "Criteo Terabyte Klicken Sie Auf Protokolle" Sind jetzt der Referenzdatensatz zur Bewertung der Skalierbarkeit VON ML-Plattformen und -Algorithmen. Mit der Prognose der Klickrate kann ein Werbetreibender die Besucher auswählen, die am ehesten auf die Werbeanzeigen antworten, seinen Browserverlauf analysieren und die relevantesten Werbeanzeigen basierend auf den Interessen des Benutzers anzeigen.

Die in diesem technischen Bericht bereitgestellte Lösung hebt die folgenden Vorteile hervor:

- Vorteile von Azure NetApp Files bei verteilten oder umfassenden Schulungen
- RAPIDS CUDA-fähige Datenverarbeitung (cuDF, cuPy usw.) und ML-Algorithmen (cuML)
- Das Parallel Computing Framework für verteilte Schulungen

Ein umfassender Workflow auf Basis VON RAPIDS AI und Azure NetApp Files zeigt eine drastische Verbesserung des Trainings für zufällige Waldmodelle im zwei Größenordnungen. Diese Verbesserung ist im Vergleich zum herkömmlichen Pandas-Ansatz signifikant, wenn es um real-world click Logs mit 45GB

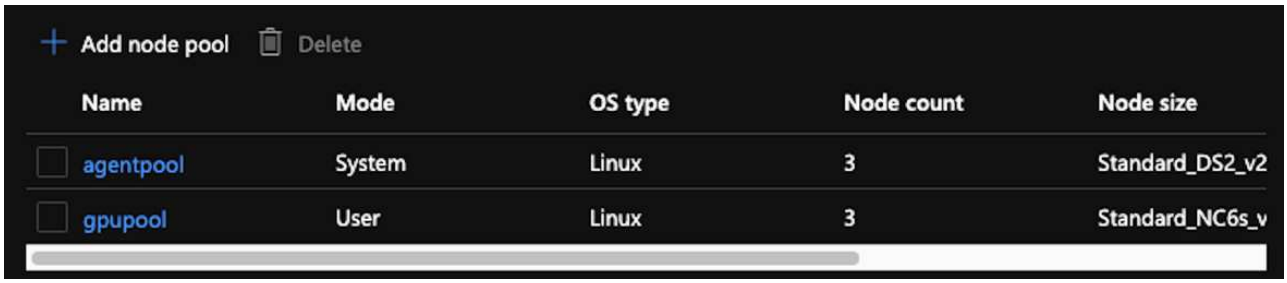
strukturierter tabellarischer Daten (im Durchschnitt) jeden Tag geht. Dies entspricht einem DataFrame mit etwa zwanzig Milliarden Zeilen. In diesem technischen Bericht werden die Einrichtung einer Cluster-Umgebung, die Installation eines Framework und einer Bibliothek, das Laden und die Verarbeitung von Daten, das konventionelle oder verteilte Training, die Visualisierung und das Monitoring demonstriert und die kritischen Ergebnisse der End-to-End-Laufzeit verglichen.

Einrichtung

Installieren und Einrichten des AKS-Clusters

Informationen zum Installieren und Einrichten des AKS-Clusters finden Sie auf der Webseite "Erstellen Sie einen AKS-Cluster" Und führen Sie dann folgende Schritte aus:

1. Wählen Sie bei Auswahl des Node-Typs (System [CPU] oder Worker [GPU] Nodes) Folgendes aus:
   a. Die Knoten des primären Systems sollten Standard DS2v2 sein (`agentpool` Standard: Drei Nodes).
   b. Fügen Sie dann den Worker-Node Standard_NC6s_v3-Pool (mindestens drei Nodes) für die Benutzergruppe (für GPU-Nodes) mit dem Namen hinzu `gpupool`.



2. Die Implementierung dauert 5 bis 10 Minuten. Klicken Sie nach Abschluss des Service auf Verbinden mit dem Cluster.
3. Um eine Verbindung zum neu erstellten AKS-Cluster herzustellen, installieren Sie Folgendes aus Ihrer lokalen Umgebung (Laptop/pc):
   a. Das Kubernetes-Befehlszeilen-Tool mithilfe des "Anweisungen für Ihr spezifisches Betriebssystem"
   b. Der Azure-CLI, wie im Dokument beschrieben, "Installieren Sie die Azure CLI"
4. Um über das Terminal auf den AKS-Cluster zuzugreifen, geben Sie ein `az login` Und geben Sie die Anmeldeinformationen ein.
5. Führen Sie die folgenden beiden Befehle aus:

```
az account set --subscription xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
aks get-credentials --resource-group resourcegroup --name aksclustername
```

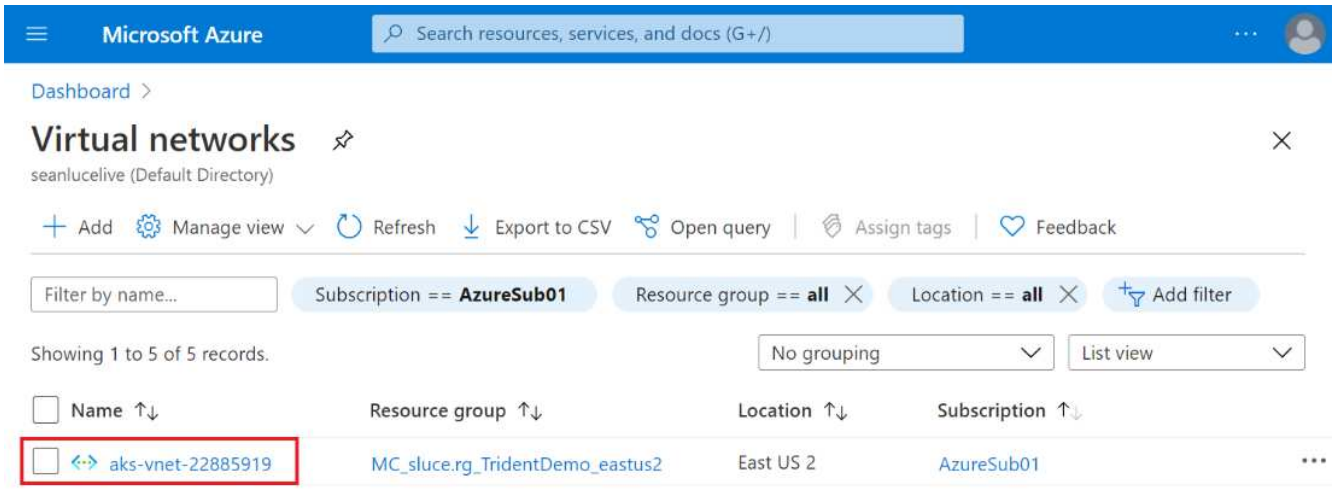
6. Eingabe `Azure CLI: kubectl get nodes.`
7. Wenn alle sechs Knoten betriebsbereit sind, wie im folgenden Beispiel dargestellt, ist Ihr AKS-Cluster bereit und mit Ihrer lokalen Umgebung verbunden

```
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 ~ %
```

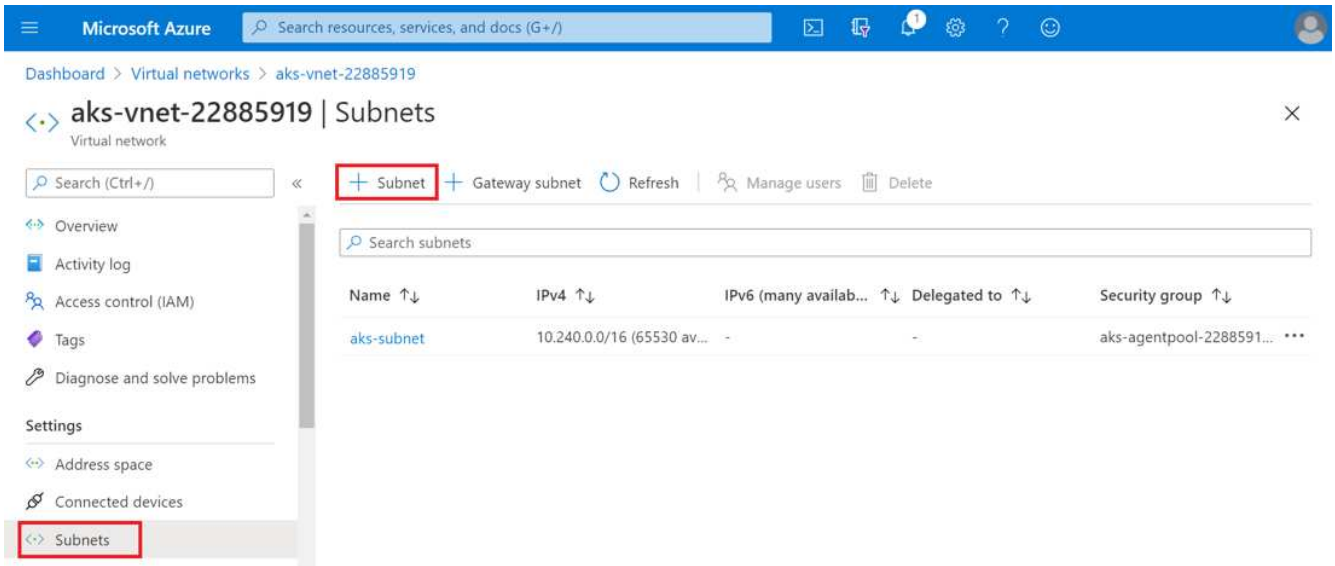
Erstellen Sie ein delegiertes Subnetz für Azure NetApp Files

Gehen Sie wie folgt vor, um ein delegiertes Subnetz für Azure NetApp Files zu erstellen:

1. Navigieren Sie im Azure-Portal zu Virtual Networks. Suchen Sie Ihr neu erstelltes virtuelles Netzwerk. Er sollte ein Präfix wie haben `aks-vnet`.
2. Klicken Sie auf den Namen des vnet.

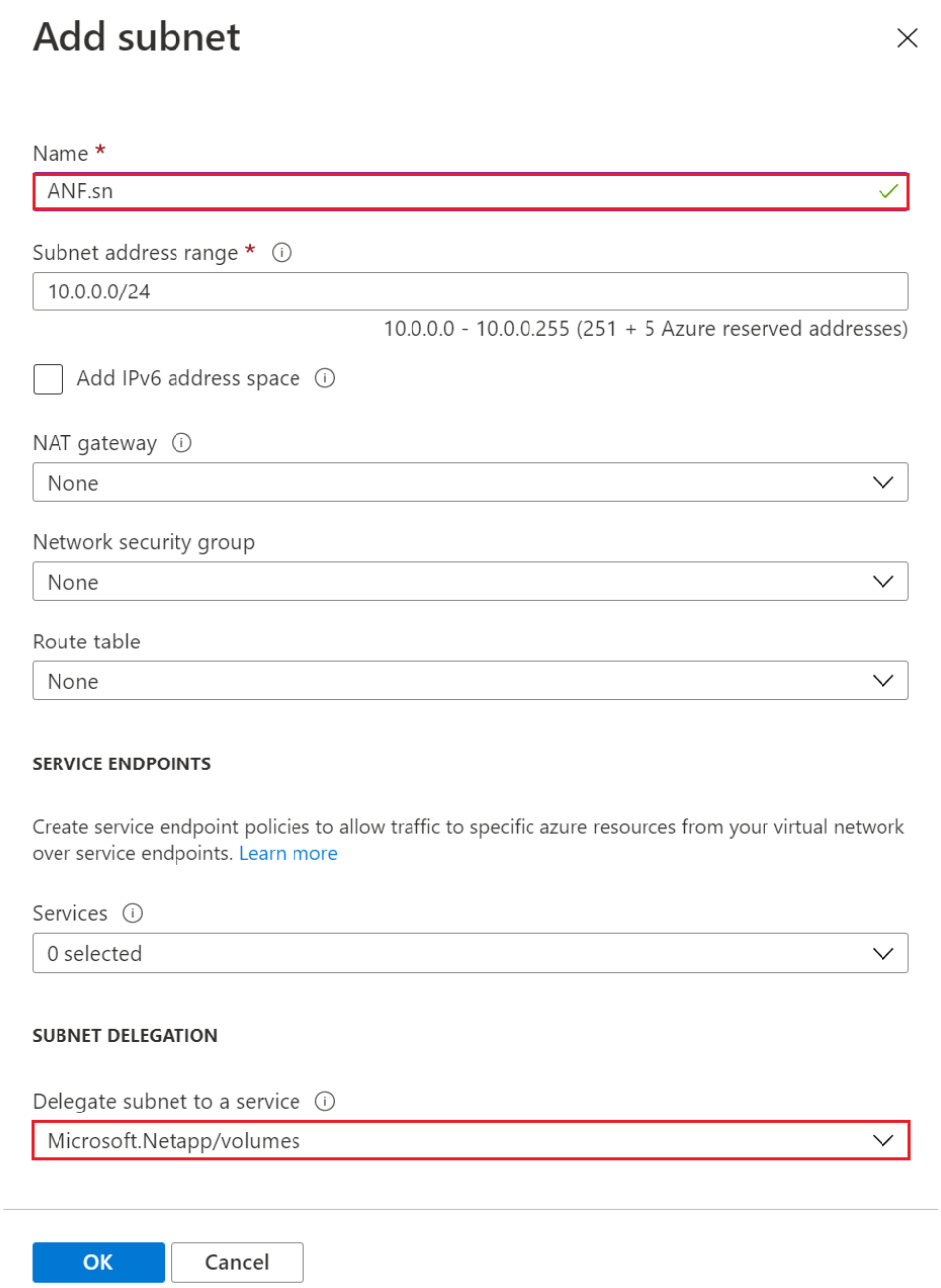


3. Klicken Sie auf Subnetze, und klicken Sie in der oberen Symbolleiste auf +Subnetz.



4. Geben Sie dem Subnetz einen Namen an, z. B. `ANF.sn` Und wählen Sie unter der Überschrift Subnet

Delegation die Option aus `Microsoft.Netapp/volumes`. Ändern Sie nichts anderes. Klicken Sie auf OK.



Azure NetApp Files Volumes werden dem Applikations-Cluster zugewiesen und als persistente Volume-Forderungen (PVCs) in Kubernetes genutzt. Dieser Prozess bietet Ihnen wiederum die Flexibilität, sie verschiedenen Services wie Jupyter Notebooks, serverlosen Funktionen usw. zuzuordnen.

Benutzer von Services können Storage auf unterschiedliche Weise von der Plattform aus nutzen. Während dieser technische Bericht über NFSs behandelt, stehen die wichtigsten Vorteile von Azure NetApp Files im Mittelpunkt:

- Benutzern die Möglichkeit zur Verwendung von Snapshot Kopien bieten
- Ermöglicht Benutzern die Speicherung großer Datenmengen auf Azure NetApp Files Volumes.
- Nutzung der Performance-Vorteile von Azure NetApp Files Volumes bei Ausführung ihrer Modelle auf

großen Dateiansätzen.

Peer AKS vnet und Azure NetApp Files vnet

Gehen Sie wie folgt vor, um das AKS-Vnet an das Azure NetApp Files-Netz anzugleichen:

1. Geben Sie im Suchfeld virtuelle Netzwerke ein.
2. Wählen Sie `vnet aks-vnet-name`. Klicken Sie darauf und geben Sie im Suchfeld Peerings ein.
3. Klicken Sie Auf + Hinzufügen.
4. Geben Sie die folgenden Deskriptoren ein:
   a. Der Peering-Linkname ist `aks-vnet-name_to_anf`.
   b. SubscriptionID und Azure NetApp Files vnet als vnet Peering-Partner.
   c. Lassen Sie alle nicht-Sternchen-Abschnitte mit den Standardwerten zurück.
5. Klicken Sie Auf Hinzufügen.

Weitere Informationen finden Sie unter "Virtuelles Netzwerk-Peering erstellen, ändern oder löschen".

Installation Von Trident

So installieren Sie Trident mithilfe von Helm:

1. Installieren Sie Helm (Installationsanweisungen finden Sie im "Quelle").
2. Laden Sie das Trident 20.01.1-Installationsprogramm herunter und extrahieren Sie es.

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

3. Ändern Sie das Verzeichnis in `trident-installer`.

```
$cd trident-installer
```

4. Kopieren `tridentctl` In ein Verzeichnis im System `$PATH`.

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

5. Installation von Trident auf dem Kubernetes (K8s) Cluster mit Helm ("Quelle"):
   a. Ändern Sie das Verzeichnis in `helm` Verzeichnis.

```
$cd helm
```

b. Installation Von Trident:

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

c. Überprüfen Sie den Status von Trident Pods.

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

Wenn alle Pods betriebsbereit sind, ist Trident installiert und Sie können weitergehen.

6. Richten Sie das Azure NetApp Files Backend und die Speicherklasse für AKS ein.

a. Erstellen eines Azure-Service-Prinzips.

Der Service-Principal ist die Art, wie Trident mit Azure kommuniziert, um die Azure NetApp Files-Ressourcen zu manipulieren.

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

Die Ausgabe sollte wie im folgenden Beispiel aussehen:

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

7. Erstellen einer json-Datei mit dem Trident-Back-End, Beispielname `anf-backend.json`.

8. Füllen Sie mithilfe Ihres bevorzugten Texteditors die folgenden Felder im aus `anf-backend.json` Datei:

```
{
    "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. Ersetzen Sie die folgenden Felder:
   - `subscriptionID`. Ihre Azure Abonnement-ID.
   - `tenantID`. Ihre Azure Mandanten-ID von der Ausgabe von `az ad sp` Im vorherigen Schritt.
   - `clientID`. Ihre appID von der Ausgabe von `az ad sp` Im vorherigen Schritt.
   - `clientSecret`. Ihr Kennwort für die Ausgabe von `az ad sp` Im vorherigen Schritt.
10. Weisen Sie Trident an, das Azure NetApp Files-Backend im zu erstellen `trident` Namespace mit `anf-backend.json` Als Konfigurationsdatei:

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

```
+------------------------+--------------------+--------------------------------------+--------+---------+
|          NAME          |   STORAGE DRIVER   |                 UUID                 | STATE  | VOLUMES |
+------------------------+--------------------+--------------------------------------+--------+---------+
| azurenetappfiles_86181 | azure-netapp-files | 2ca85462-59ac-4946-be05-c03f5575a2ad | online |       0 |
+------------------------+--------------------+--------------------------------------+--------+---------+
```

11. Erstellen Sie eine Speicherklasse. Kubernetes Benutzer stellen Volumes mithilfe von PVCs bereit, die eine Storage-Klasse nach Namen angeben. Weisen Sie K8s an, eine Speicherklasse zu erstellen `azurenetappfiles` Damit wird auf das im vorherigen Schritt erstellte Trident Back-End Bezug gezogen.
12. Erstellen Sie ein YAML (`anf-storage-class.yaml`) Datei für Speicherklasse und kopieren.

```
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. Überprüfen Sie, ob die Speicherklasse erstellt wurde.

```
kubectl get sc azurenetappfiles
```

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

Einrichten von „Dask with RAPIDS Deployment“ auf AKS mit Helm

Gehen Sie wie folgt vor, um die Implementierung von „Dask with RAPIDS“ auf AKS mit Helm einzurichten:

1. Erstellen Sie einen Namespace für die Installation von DASK mit RAPIDS.

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

2. Erstellen Sie ein PVC, um den Click-Through-Rate-Datensatz zu speichern:
   a. Speichern Sie den folgenden YAML-Inhalt in einer Datei, um ein PVC zu erstellen.

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

   b. Wenden Sie die YAML-Datei auf Ihr Kubernetes-Cluster an.

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

3. Klonen Sie die `rapidsai git` Repository ("https://github.com/rapidsai/helm-chart").

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

4. Ändern `values.yaml` Und schließen Sie die PVC ein, die früher für Arbeiter und Jupyter-Workspace erstellt wurde.
   a. Wechseln Sie zum `rapidsai` Verzeichnis des Repository.

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

   b. Aktualisieren Sie die `values.yaml` Datei und Mounten des Volumes mit PVC.

```
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. Gehen Sie zum Home-Verzeichnis des Projektarchivs und stellen Sie Dask mit drei Worker-Knoten auf

AKS mit Helm bereit.

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

Azure NetApp Files-Performance-Tiers

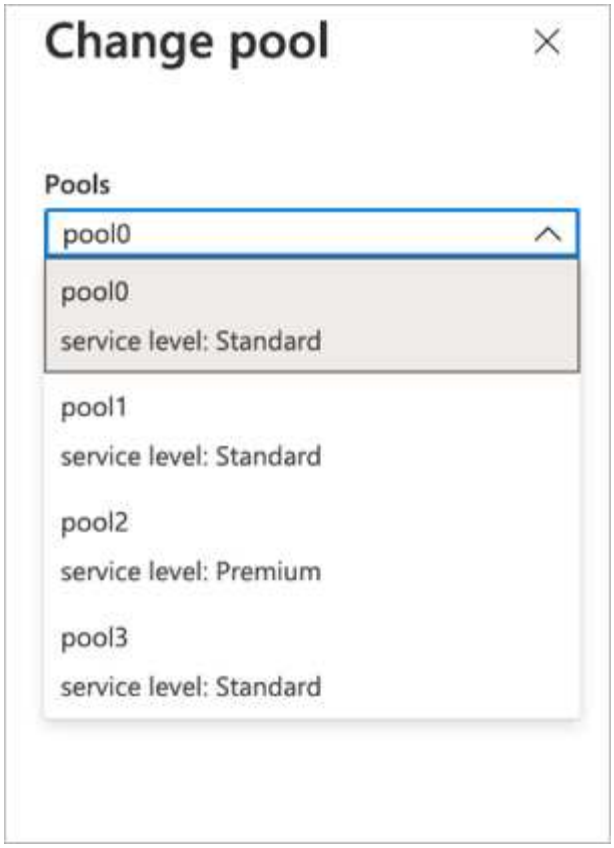
Sie können den Service-Level eines vorhandenen Volumes ändern, indem Sie das Volume in einen anderen Kapazitäts-Pool verschieben und dabei das gewünschte Service-Level für das Volume nutzen. Mithilfe dieser Lösung können Kunden mit einem kleinen Datensatz und einer kleinen Anzahl GPUs in der Standard-Ebene beginnen und bei zunehmender Datenmenge und GPUs horizontal oder vertikal auf Premium-Tier skalieren. Die Premium-Ebene bietet als Standardebene einen viermal so hohen Durchsatz pro Terabyte. Die vertikale Skalierung erfolgt ohne dass Daten verschoben werden müssen, um den Service Level eines Volumes zu ändern.

Profitieren Sie von einer dynamischen Änderung des Service-Levels eines Volumes

Um den Service-Level eines Volumes dynamisch zu ändern, gehen Sie wie folgt vor:

1. Klicken Sie auf der Seite Volumes mit der rechten Maustaste auf das Volume, dessen Service-Level Sie ändern möchten. Wählen Sie Pool Ändern.

2. Wählen Sie im Fenster Pool ändern den Kapazitätspool aus, in den Sie das Volume verschieben möchten.



3. Klicken Sie auf OK.

Automatisieren Sie Änderungen in Performance-Tiers

Zur Automatisierung von Änderungen in der Performance-Ebene stehen folgende Optionen zur Verfügung:

- Die dynamische Änderung des Service-Levels befindet sich derzeit noch in der öffentlichen Vorschau und ist standardmäßig nicht aktiviert. Um diese Funktion im Azure-Abonnement zu aktivieren, lesen Sie in dieser Dokumentation nach, wie Sie diese Funktion aktivieren können "Profitieren Sie von einer dynamischen Änderung des Service-Levels eines Volumes".
- In werden Änderungsbefehle für den Azure CLI-Volume-Pool bereitgestellt "Dokumentation der Änderungen am Volume Pool" Und im folgenden Beispiel:

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

- PowerShell: Das "Cmdlet "AzNetAppFilesVolumePool" Ändert den Pool eines Azure NetApp Files Volume und wird im folgenden Beispiel dargestellt:

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

Klicken Sie sich durch die Verarbeitung der Tarifprognosen und das Modelltraining

Bibliotheken für die Datenverarbeitung und das Modelltraining

In der folgenden Tabelle sind die Bibliotheken und Frameworks aufgeführt, die zum Erstellen dieser Aufgabe verwendet wurden. Alle diese Komponenten wurden vollständig in die rollenbasierte Zugriffs- und Sicherheitskontrolle von Azure integriert.

Bibliotheken/Framework	Beschreibung
Fragen Sie die CuML	Für ML zur Arbeit an GPU, das "CuML-Bibliothek" Bietet Zugriff auf DAS RAPIDS CuML-Paket mit DASK. RAPIDS CuML implementiert gängige ML-Algorithmen wie Clustering, Dimensionierungsreduzierung und Regression. Hochperformante GPU-basierte Implementierungen ermöglichen eine bis zu 100-fache Geschwindigkeit gegenüber CPU-basierten Ansätzen.
Fragen Sie nach cuDF	CuDF umfasst verschiedene andere Funktionen, die GPU-beschleunigte Extraktion, Transformation, Last (ETL) unterstützen, wie z. B. Untersetzung von Daten, Transformationen, One-Hot-Codierung und mehr. Das Team VON RAPIDS unterhält eine "Dask-cudf Bibliothek" Dazu gehören Hilfsmethoden für die Verwendung von Fragen und CuDF.
Ischikit Lernen	Scikit-Learn bietet Dutzende von integrierten Algorithmen und Modellen für maschinelles Lernen an, die als Schätzer bezeichnet werden. Beide "kostenplaner" Kann mit dem an einige Daten angepasst werden "Fit" Methode.

Zum Vergleich haben wir zwei Notebooks für DIE ERSTELLUNG der ML-Pipelines eingesetzt, eines ist der konventionelle Pandas-Scikit-Learn-Ansatz, und das andere ist Distributed Training mit RAPIDS und Dask. Jedes Notebook kann einzeln getestet werden, um die Leistung in Bezug auf Zeit und Umfang zu sehen. Wir decken jedes Notebook einzeln ab, um die Vorteile des Distributed Trainings mit RAPIDS und Dask zu demonstrieren.

Criteo Click Logs Tag 15 in Pandas laden und ein scikit-Learn Zufallswaldmodell trainieren

In diesem Abschnitt wird beschrieben, wie wir Pandas und DataFrames zum Laden von Click Logs-Daten aus dem Criteo Terabyte-Datensatz verwendet haben. Der Anwendungsfall ist in der digitalen Werbung relevant, damit Anzeigenaustausch Nutzer-Profile bauen kann, indem er vorhersagt, ob Werbeanzeigen angeklickt werden oder wenn der Austausch kein genaues Modell in einer automatisierten Pipeline verwendet.

Wir haben Tag 15 Daten aus dem Click Logs Datensatz geladen, insgesamt 45GB. Ausführen der folgenden Zelle im Jupyter-Notebook `CTR-PandasRF-collated.ipynb` Erstellt einen Pandas DataFrame, der die ersten 50 Millionen Zeilen enthält und ein scikit-Learn Zufallswaldmodell erzeugt.

```
%%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)
```

Führen Sie den folgenden Abschnitt in diesem Notizbuch aus, um eine Vorhersage mit einem trainierten zufälligen Waldmodell durchzuführen. Wir haben die letzten 1 Million Zeilen ab Tag 15 als Testsatz genommen, um mögliche Duplikate zu vermeiden. Die Zelle berechnet auch die Genauigkeit der Vorhersage, definiert als Prozentsatz der Vorkommen das Modell sagt genau aus, ob ein Benutzer auf eine Anzeige klickt oder nicht. Informationen zu unbekannten Komponenten in diesem Notebook finden Sie im "Offizielle Scikit-Learn-Dokumentation".

```
# 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('col1')
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))
```

Laden Sie Tag 15 in Damast und trainieren Sie ein Damast CuML zufälligen Wald Modell

Ähnlich wie im vorherigen Abschnitt, laden Criteo Click Logs Tag 15 in Pandas und trainieren ein scikit-Learn Zufallswaldmodell. In diesem Beispiel haben wir DataFrame-Beladung mit Dask cuDF durchgeführt und ein zufälliges Waldmodell in Dask cuML trainiert. Im Abschnitt wurden die Unterschiede in Trainingszeit und Umfang verglichen "„Zeitvergleich Training“."

criteo_dask_RF.ipynb

Dieses Notizbuch importiert `numpy`, `cuml`, Und das nötige `dask` Bibliotheken, wie im folgenden Beispiel gezeigt:

```
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
```

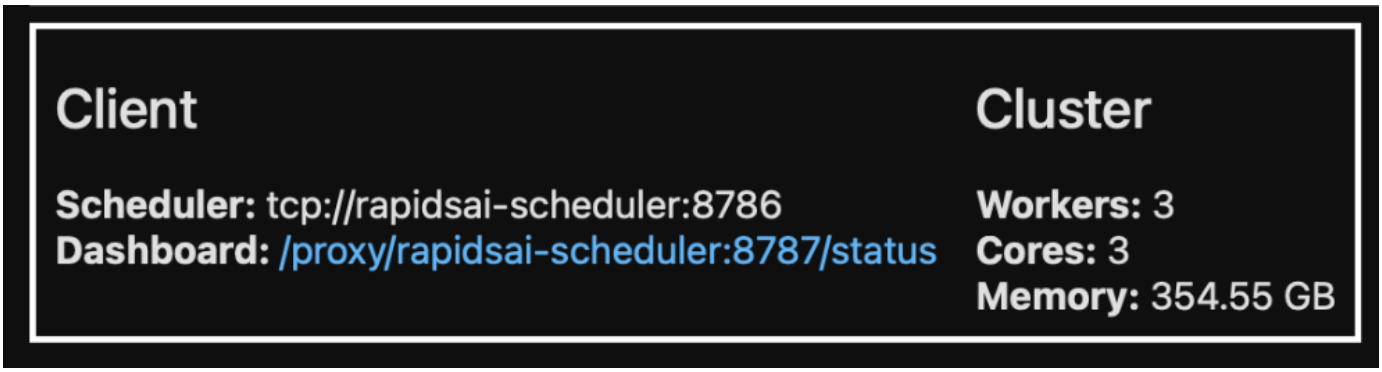
Dask Client() Initiieren.

```
client = Client()
```

Wenn das Cluster ordnungsgemäß konfiguriert ist, können Sie den Status von „Worker“-Nodes anzeigen.

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

In unserem AKS-Cluster wird folgender Status angezeigt:



Beachten Sie, dass Dask das faule Ausführungsmodell verwendet: Anstatt den Verarbeitungscode sofort auszuführen, erstellt Dask stattdessen ein gesteuertes Acyclic Graph (DAG) der Ausführung. DAG umfasst eine Reihe von Aufgaben und ihre Interaktionen, die jeder Mitarbeiter ausführen muss. Dieses Layout bedeutet, dass die Tasks nicht ausgeführt werden, bis der Benutzer DASK anweist, sie auf die eine oder andere Weise auszuführen. Mit DASK haben Sie drei Hauptoptionen:

- **Aufruf Compute() auf einem DataFrame.** dieser Aufruf verarbeitet alle Partitionen und gibt die Ergebnisse dann in den Scheduler für die abschließende Aggregation und Konvertierung in cuDF DataFrame zurück. Diese Option sollte sparsam und nur bei stark reduzierten Ergebnissen verwendet werden, es sei denn, Ihr Scheduler-Node verfügt über keinen Arbeitsspeicher.
- **Call persist() auf einem DataFrame.** dieser Aufruf führt die Grafik aus, aber anstatt die Ergebnisse an den Scheduler-Knoten zurückzugeben, wird sie über den Cluster im Speicher verwaltet, sodass der Benutzer diese Zwischenergebnisse in der Pipeline wiederverwenden kann, ohne dass eine erneute Verarbeitung erforderlich ist.
- **Call Head() auf einem DataFrame.** wie bei cuDF gibt dieser Aufruf 10 Datensätze zurück zum Scheduler-Knoten. Mit dieser Option können Sie schnell überprüfen, ob Ihr DataFrame das gewünschte Ausgabeformat enthält oder ob die Datensätze selbst in Abhängigkeit von Ihrer Verarbeitung und Berechnung sinnvoll sind.

Wenn der Benutzer eine dieser Aktionen nicht anruft, warten die Arbeiter daher nicht, bis der Planer die Verarbeitung initiiert hat. Dieses faul Ausführungsparadigma ist häufig in modernen Parallel- und Distributed Computing Frameworks wie Apache Spark zu finden.

Im folgenden Abschnitt wird ein Zufallswaldmodell mit Hilfe von Dask cuML für verteiltes GPU-beschleunigtes Computing trainiert und die Genauigkeit der Modellvorhersage berechnet.

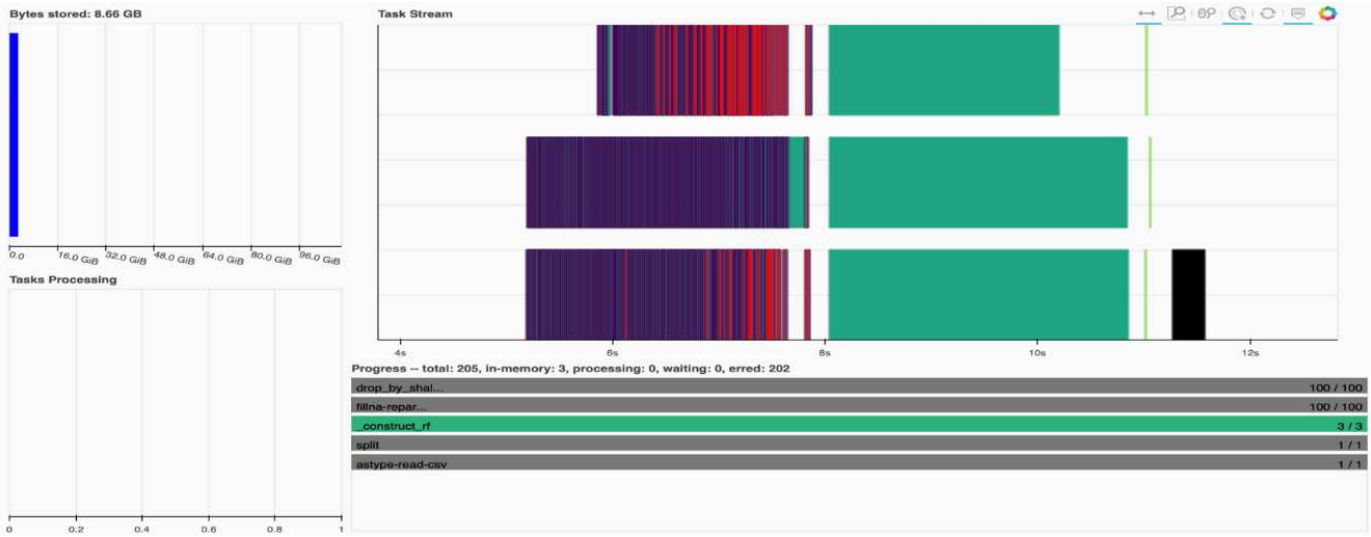
```
Adsf
# Random Forest building parameters
n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
cuml_model = cumlDaskRF(max_depth=max_depth, n_estimators=n_trees,
n_bins=n_bins, n_streams=n_streams, verbose=True, client=client)
cuml_model.fit(gdf_sliced_small, Y)
# Model prediction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

Überwachung des DASK mit dem systemeigenen Task-Streams-Dashboard

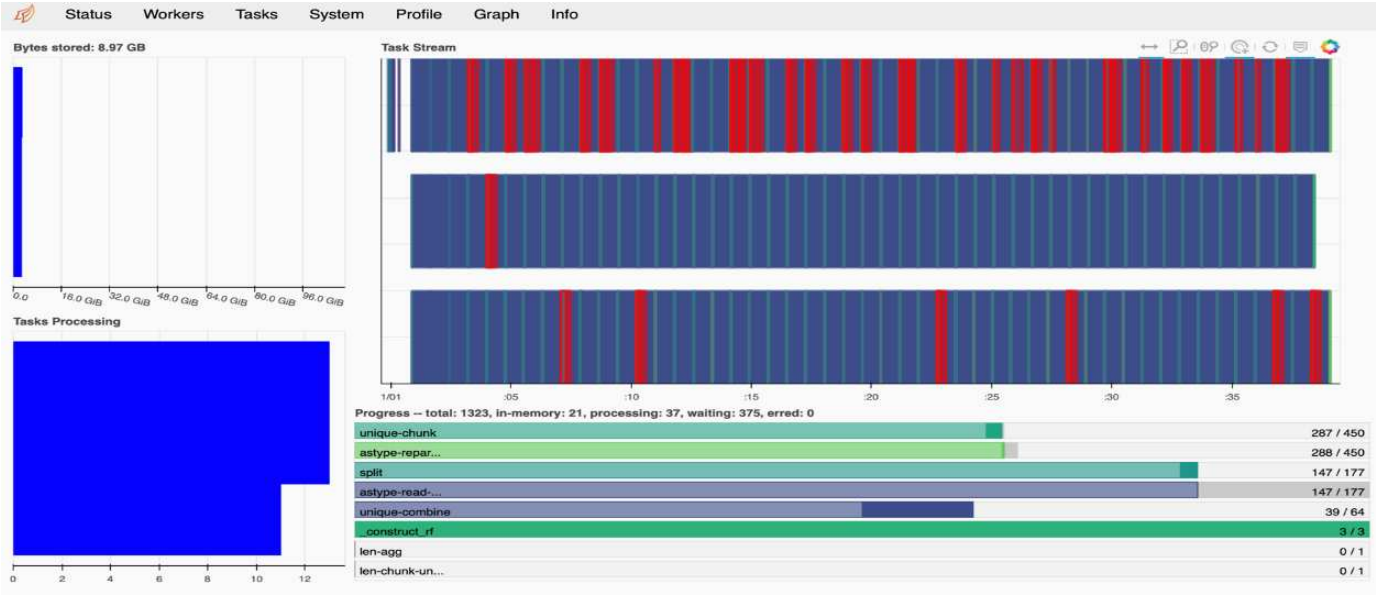
Der "Distributed Scheduler abfragen" Live-Feedback in zwei Formen:

- Ein interaktives Dashboard mit vielen Darstellungen und Tabellen mit Live-Informationen
- Eine Fortschrittsleiste, die für den interaktiven Einsatz in Konsolen oder Notebooks geeignet ist

In unserem Fall zeigt die folgende Abbildung, wie Sie den Fortschritt der Aufgabe überwachen können, einschließlich gespeicherter Bytes, den Task-Stream mit einer detaillierten Aufschlüsselung der Anzahl von Streams und den Fortschritt nach Aufgabennamen mit den ausgeführten Funktionen. In unserem Fall, weil wir drei Arbeiter-Knoten haben, gibt es drei Hauptblöcke des Streams und die Farbcodes bezeichnen verschiedene Aufgaben innerhalb jedes Streams.



Sie haben die Möglichkeit, einzelne Aufgaben zu analysieren und die Ausführungszeit in Millisekunden zu untersuchen oder Hindernisse oder Hindernisse zu identifizieren. Die folgende Abbildung zeigt beispielsweise die Aufgabenströme für die Zufallsphase des Waldmodells. Es werden wesentlich mehr Funktionen ausgeführt, darunter ein einzigartiger Chunk für die DataFrame-Verarbeitung, _construct_rf für die Anpassung der zufälligen Gesamtstruktur usw. Die meiste Zeit wurde auf DataFrame-Operationen aufgrund der großen Größe (45 GB) von einem Tag Daten aus den Criteo Click Logs verwendet.



Vergleich der Schulungszeit

Dieser Abschnitt vergleicht die Modelltrainings-Zeit mit konventionellem Pandas im Vergleich zu Dask. Bei Pandas haben wir eine kleinere Menge an Daten geladen, weil die Verarbeitungszeit langsamer ist, um Speicherüberlauf zu vermeiden. Daher haben wir die Ergebnisse interpoliert, um einen fairen Vergleich zu bieten.

Die folgende Tabelle zeigt den Vergleich der Rohtrainings-Zeiten, wenn für das Modell der zufälligen Wälder von Pandas deutlich weniger Daten benötigt werden (50 Millionen Zeilen von 20 Milliarden pro Tag 15 des Datensatzes). Diese Stichprobe benötigt nur weniger als 0.25 % aller verfügbaren Daten. Während wir für Dask-cuML das Zufallswaldmodell auf allen 20 Milliarden verfügbaren Reihen trainiert haben. Die beiden Ansätze ergaben eine vergleichbare Trainingszeit.

Vorgehensweise	Schulungszeit
Scikit-Learn: Verwendung von nur 50M Reihen im Tag15 als Trainingsdaten	47 Minuten und 21 Sekunden
RAPIDS-DASK: Alle 20B-Reihen täglich an15 als Trainingsdaten nutzen	1 Stunde, 12 Minuten und 11 Sekunden

Wenn wir die Ergebnisse der Trainingszeit linear interpolieren, wie in der folgenden Tabelle dargestellt, bietet sich ein bedeutender Vorteil für die Verwendung verteilter Schulungen mit Dask. Es würde 13 Tage dauern, bis der konventionelle Pandas scikit-Lernansatz 45GB Daten für einen einzigen Tag mit Klick-Protokollen verarbeitet und trainiert, während der RAPIDS-DASK-Ansatz die gleiche Datenmenge 262.39-mal schneller verarbeitet.

Vorgehensweise	Schulungszeit
Scikit-Learn: Alle 20B-Reihen im Tag15 als Trainingsdaten verwenden	13 Tage, 3 Stunden, 40 Minuten und 11 Sekunden
RAPIDS-DASK: Alle 20B-Reihen täglich an15 als Trainingsdaten nutzen	1 Stunde, 12 Minuten und 11 Sekunden

In der vorherigen Tabelle ist zu sehen, dass durch die Verwendung VON RAPIDS mit DASK die

Datenverarbeitung und das Modelltraining über mehrere GPU-Instanzen hinweg deutlich kürzer ist als bei der konventionellen Pandas DataFrame-Verarbeitung mit scikit-Learn Modelltrainings. Dieses Framework ermöglicht sowohl vertikale als auch horizontale Skalierung in der Cloud als auch On-Premises in einem Multi-Node-Cluster mit mehreren GPUs.

Überwachen Sie „Dask and RAPIDS“ mit Prometheus und Grafana

Nutzen Sie nach der Implementierung alles Rückschlüsse auf neue Daten. Die Modelle prognostizieren, ob ein Benutzer basierend auf Browsing-Aktivitäten auf eine Anzeige klickt. Die Ergebnisse der Vorhersage werden in einem Damask cuDF gespeichert. Sie können die Ergebnisse mit Prometheus überwachen und in Grafana Dashboards visualisieren.

Weitere Informationen finden Sie unter "RAPIDS AI Medium Post".

Datensatz- und Modellversionierung mit NetApp DataOps Toolkit

Das NetApp DataOps Toolkit für Kubernetes abstrahiert Storage-Ressourcen und Kubernetes-Workloads bis auf die Ebene des Data Science Workspace. Diese Funktionen sind in einer einfachen, benutzerfreundlichen Oberfläche zusammengestellt, die für Data Scientists und Data Engineers konzipiert wurde. Mit der bekannten Form eines Python-Programms ermöglicht das Toolkit Data Scientists und Ingenieuren die Bereitstellung und Zerstörung von JupyterLab-Arbeitsbereichen in nur wenigen Sekunden. Diese Workspaces können Storage-Kapazität im Terabyte- oder sogar Petabyte-Bereich enthalten. Data Scientists können all ihre Trainingsdatensätze direkt in ihren Projektarbeitsbereichen speichern. Die Tage für das separate Management von Workspaces und Daten-Volumes sind vorbei.

Weitere Informationen finden Sie im Toolkit "GitHub Repository".

Jupyter-Notebooks als Referenz

Diesem technischen Bericht sind zwei Jupyter Notebooks zugeordnet:

- **"CTR-PandasRF-collated.ipynb."** Dieses Notebook lädt Tag 15 aus dem Criteo Terabyte Click Protokolldatensatz, verarbeitet und formatiert Daten in einen Pandas DataFrame, trainiert ein Scikit-Learn Zufallswaldmodell, führt Vorhersage aus und berechnet Genauigkeit.
- **"criteo_dask_RF.ipynb."** Dieses Notebook lädt Tag 15 aus dem Criteo Terabyte Click Log Datensatz, verarbeitet und formatiert Daten in einen Damast CuDF, trainiert ein Dusk CuML Zufallswaldmodell, führt Vorhersage aus und berechnet Genauigkeit. Durch die Nutzung von mehreren Worker-Nodes mit GPUs ist dieser verteilte Daten- und Modellverarbeitungs- und Trainingsansatz äußerst effizient. Je mehr Daten Sie verarbeiten, desto größer ist die Zeitersparnis im Vergleich zu einem herkömmlichen ML-Ansatz. Dieses Notebook lässt sich in der Cloud, vor Ort oder in einer hybriden Umgebung bereitstellen, in der Ihr Kubernetes-Cluster Computing und Storage an verschiedenen Standorten enthält, sofern Ihr Netzwerk-Setup Daten und die Modellverteilung frei ermöglicht.

Schlussfolgerung

Azure NetApp Files, RAPIDS und Dask beschleunigen und vereinfachen die

Implementierung umfangreicher ML-Verarbeitung und -Trainings durch die Integration mit Orchestrierungs-Tools wie Docker und Kubernetes. Durch die Vereinheitlichung der End-to-End-Datenpipeline verringert diese Lösung die Latenz und Komplexität vieler moderner Computing-Workloads und schließt damit die Lücke zwischen Entwicklung und Betrieb. Data Scientists können Abfragen zu großen Datensätzen durchführen und Daten und algorithmische Modelle während der Trainingsphase sicher mit anderen Benutzern teilen.

Bei der Erstellung eigener KI/ML-Pipelines ist die Konfiguration der Integration, Verwaltung, Sicherheit und Zugänglichkeit der Komponenten einer Architektur eine schwierige Aufgabe. Der Zugriff auf und die Kontrolle über ihre Umgebung für Entwickler stellt eine weitere Herausforderung dar.

Durch die Entwicklung eines End-to-End-Trainingsmodells und einer verteilten Datenpipeline in der Cloud haben wir gegenüber einem herkömmlichen Open-Source-Ansatz zwei Möglichkeiten zur Verbesserung der gesamten Workflow-Vervollständigung aufgezeigt, der die GPU-beschleunigte Datenverarbeitung und Computing Frameworks nicht ausübte.

Die Kombination aus NetApp, Microsoft, opensource-Orchestrierungs-Frameworks und NVIDIA vereint die neuesten Technologien als Managed Services mit hoher Flexibilität und ermöglicht so eine beschleunigte Technologieeinführung und eine kürzere Markteinführungszeit für neue KI/ML-Applikationen. Diese erweiterten Services werden in einer Cloud-nativen Umgebung bereitgestellt, die sowohl für On-Premises- als auch für hybride Architekturen problemlos portiert werden kann.

Wo Sie weitere Informationen finden

Folgende Ressourcen enthalten ausführlichere Informationen zu den Angaben in diesem Dokument:

- Azure NetApp Files
  - Lösungsseite zur Architektur für Azure NetApp Files

    "https://docs.microsoft.com/azure/azure-netapp-files/azure-netapp-files-solution-architectures"
- Persistenter Trident Storage für Container:
  - Azure NetApp Files und Trident zu überzeugen

    "https://netapptrident.readthedocs.io/en/stablev20.07/kubernetes/operations/tasks/backends/anf.html"
- Dask und RAPIDS:
  - Fragen Sie Nach

    "https://docs.dask.org/en/latest/"
  - Installieren Sie Die Dask

    "https://docs.dask.org/en/latest/install.html"
  - DASK-API

    "https://docs.dask.org/en/latest/api.html"

    - Maschinelles Lernen Absprechen

      "https://examples.dask.org/machine-learning.html"

    - Distributed Diagnostics Abfragen

      "https://docs.dask.org/en/latest/diagnostics-distributed.html"

- ML-Rahmen und Werkzeuge:
    - TensorFlow: Ein Open-Source Machine Learning Framework für alle

      "https://www.tensorflow.org/"

    - Docker

      "https://docs.docker.com"

    - Kubernetes

      "https://kubernetes.io/docs/home/"

    - Kubeflow

      "http://www.kubeflow.org/"

    - Jupyter Notebook Server

      "http://www.jupyter.org/"

Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGENDEINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter http://www.netapp.com/TM aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.