Sistema operativo ai per il cloud ibrido con caching dei dati

NetApp Solutions

NetApp
April 26, 2024

Sommario

TR-4841: Sistema operativo ai per il cloud ibrido con caching dei dati

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

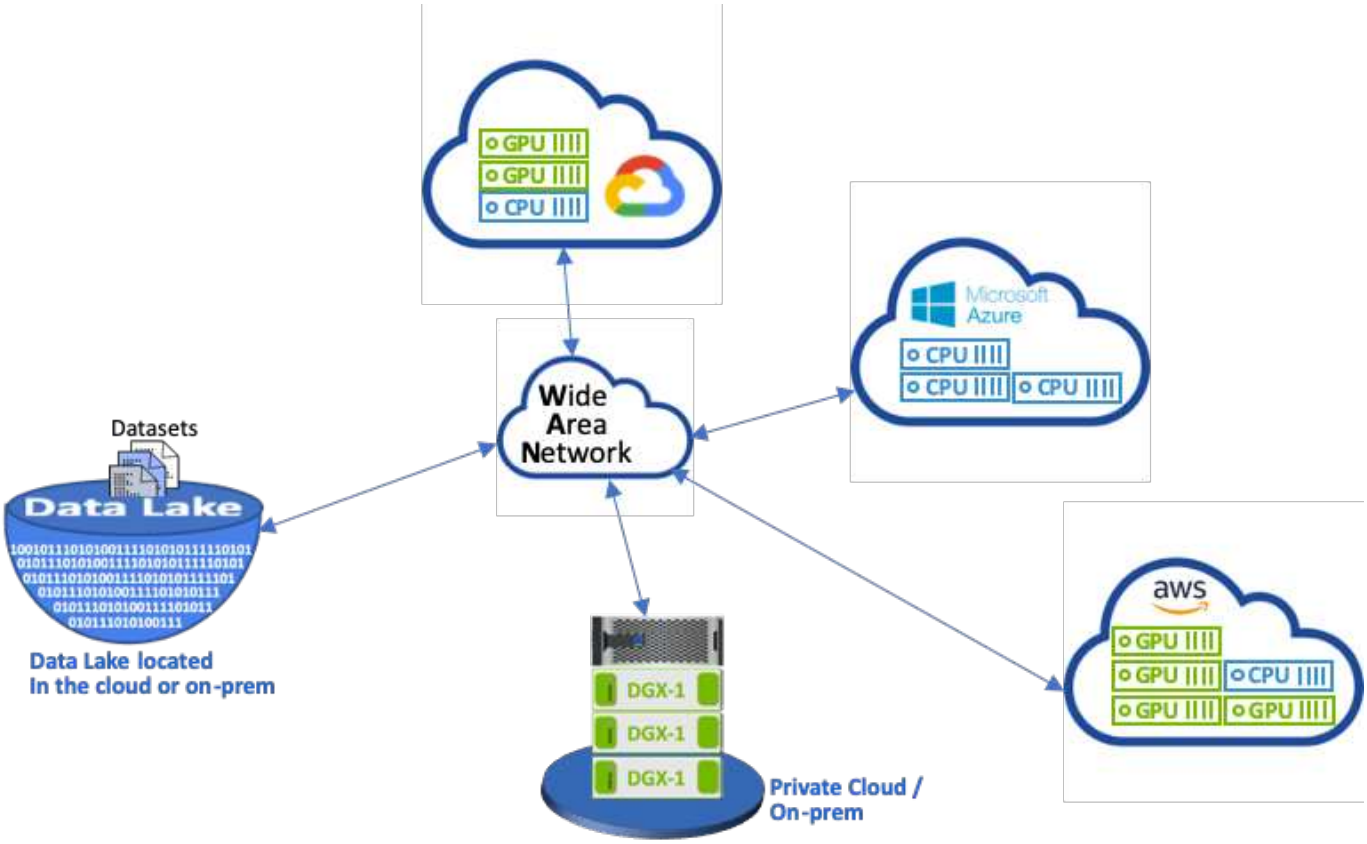
La crescita esplosiva dei dati e la crescita esponenziale di ML e ai sono convergenti per creare un'economia di zettabyte con sfide di sviluppo e implementazione uniche.

Sebbene sia noto che i modelli ML sono affamati di dati e richiedono uno storage dei dati ad alte performance prossimale alle risorse di calcolo, in pratica non è così semplice implementare questo modello, soprattutto con il cloud ibrido e le istanze di calcolo elastiche. In genere, enormi quantità di dati vengono memorizzate in data Lake a basso costo, dove le risorse di calcolo ai dalle performance elevate, come le GPU, non possono accedervi in modo efficiente. Questo problema è aggravato in un'infrastruttura di cloud ibrido in cui alcuni carichi di lavoro operano nel cloud e alcuni si trovano on-premise o in un ambiente HPC completamente diverso.

In questo documento, presentiamo una nuova soluzione che consente ai professionisti IT e ai data engineer di creare una piattaforma di cloud ai realmente ibrido con un data hub consapevole della topologia che consente ai data scientist di creare istantaneamente e automaticamente una cache dei propri set di dati in prossimità delle proprie risorse di calcolo. ovunque si trovino. Di conseguenza, non solo è possibile ottenere un training con modelli ad alte performance, ma si creano anche ulteriori benefici, tra cui la collaborazione di diversi professionisti dell'ai, che hanno accesso immediato a cache, versioni e linee di dati all'interno di un hub di versione del set di dati.

Panoramica del caso d'utilizzo e Problem Statement

I set di dati e le versioni dei set di dati si trovano in genere in un data Lake, come lo storage basato su oggetti NetApp StorageGRID, che offre costi ridotti e altri vantaggi operativi. Gli scienziati dei dati estraggono questi set di dati e li progettano in più fasi per prepararli alla formazione con un modello specifico, spesso creando più versioni lungo il percorso. Come fase successiva, il data scientist deve scegliere risorse di calcolo ottimizzate (GPU, istanze di CPU high-end, un cluster on-premise e così via) per eseguire il modello. La figura seguente mostra la mancanza di prossimità del dataset in un ambiente di calcolo ML.



Tuttavia, è necessario eseguire più esperimenti di training in parallelo in diversi ambienti di calcolo, ciascuno dei quali richiede il download del dataset dal data Lake, un processo costoso e lungo. La prossimità del set di dati all'ambiente di calcolo (in particolare per un cloud ibrido) non è garantita. Inoltre, gli altri membri del team che eseguono i propri esperimenti con lo stesso set di dati devono eseguire lo stesso arduo processo. Al di là dell'evidente rallentamento dell'accesso ai dati, le sfide includono il monitoraggio delle versioni dei set di dati, la condivisione dei set di dati, la collaborazione e la riproducibilità.

Requisiti del cliente

I requisiti dei clienti possono variare per ottenere esecuzioni ML dalle performance elevate utilizzando le risorse in modo efficiente; ad esempio, i clienti potrebbero richiedere quanto segue:

- Accesso rapido ai set di dati da ogni istanza di calcolo che esegue il modello di training senza incorrere in costose complessità di download e accesso ai dati
- L'utilizzo di qualsiasi istanza di calcolo (GPU o CPU) nel cloud o on-premise senza preoccuparsi della posizione dei set di dati
- Maggiore efficienza e produttività grazie all'esecuzione di più esperimenti di training in parallelo con diverse risorse di calcolo sullo stesso set di dati senza ritardi e latenza dei dati non necessari
- Costi delle istanze di calcolo ridotti al minimo
- Riproducibilità migliorata grazie a tool per la conservazione dei record dei set di dati, della loro discendenza, delle versioni e di altri dettagli sui metadati
- Condivisione e collaborazione migliorate per consentire a qualsiasi membro autorizzato del team di accedere ai set di dati ed eseguire esperimenti

Per implementare il caching dei set di dati con il software per la gestione dei dati NetApp ONTAP, i clienti devono eseguire le seguenti attività:

- Configurare e impostare lo storage NFS più vicino alle risorse di calcolo.
- Determinare il set di dati e la versione da memorizzare nella cache.
- Monitorare la memoria totale impegnata nei set di dati memorizzati nella cache e la quantità di storage NFS disponibile per ulteriori commit di cache (ad esempio, gestione della cache).
- Esaurire i set di dati nella cache se non sono stati utilizzati in un determinato periodo di tempo. L'impostazione predefinita è un giorno; sono disponibili altre opzioni di configurazione.

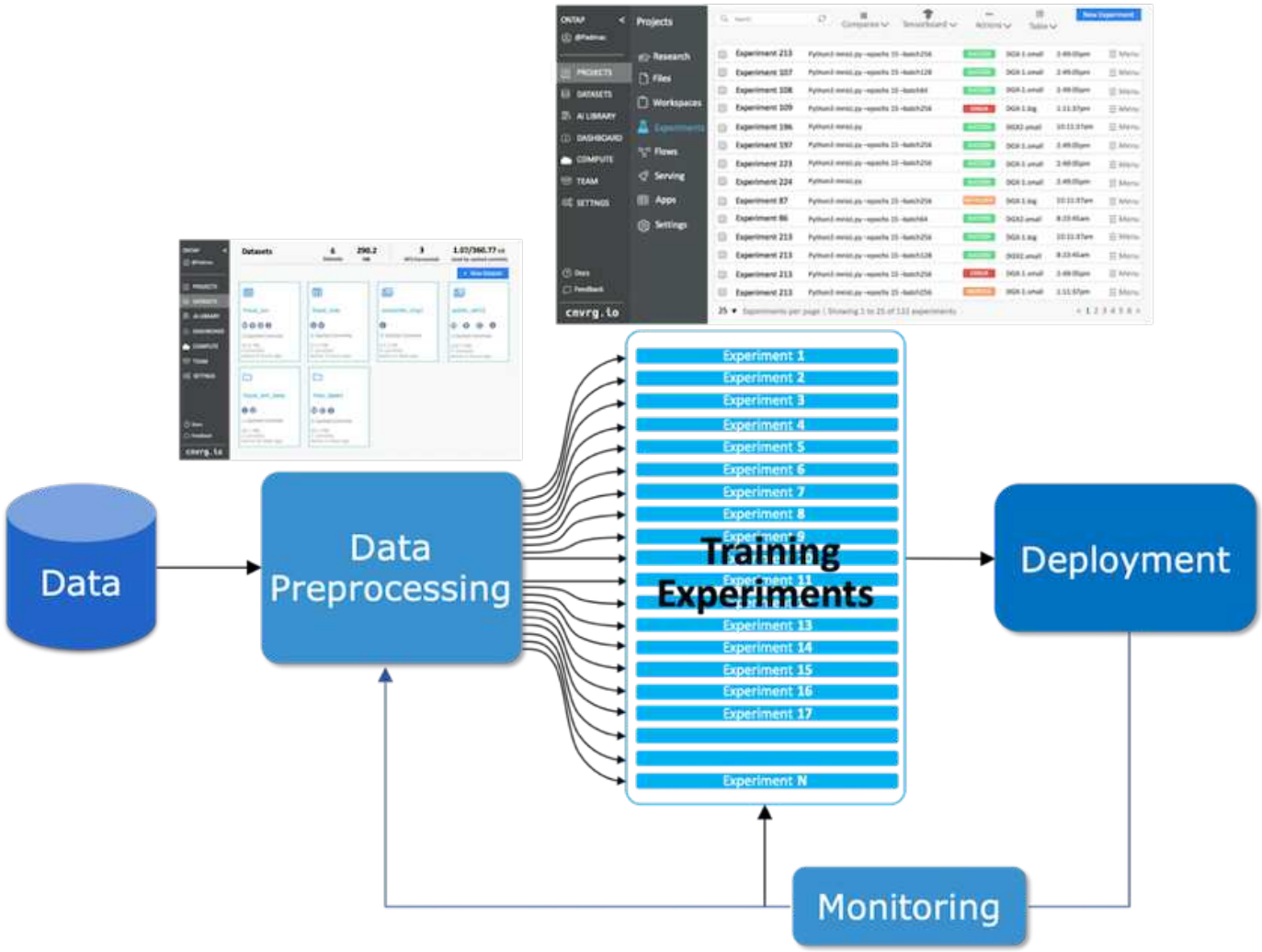
Panoramica della soluzione

In questa sezione viene descritta una pipeline convenzionale per la scienza dei dati e i relativi inconvenienti. Presenta inoltre l'architettura della soluzione di caching dei set di dati proposta.

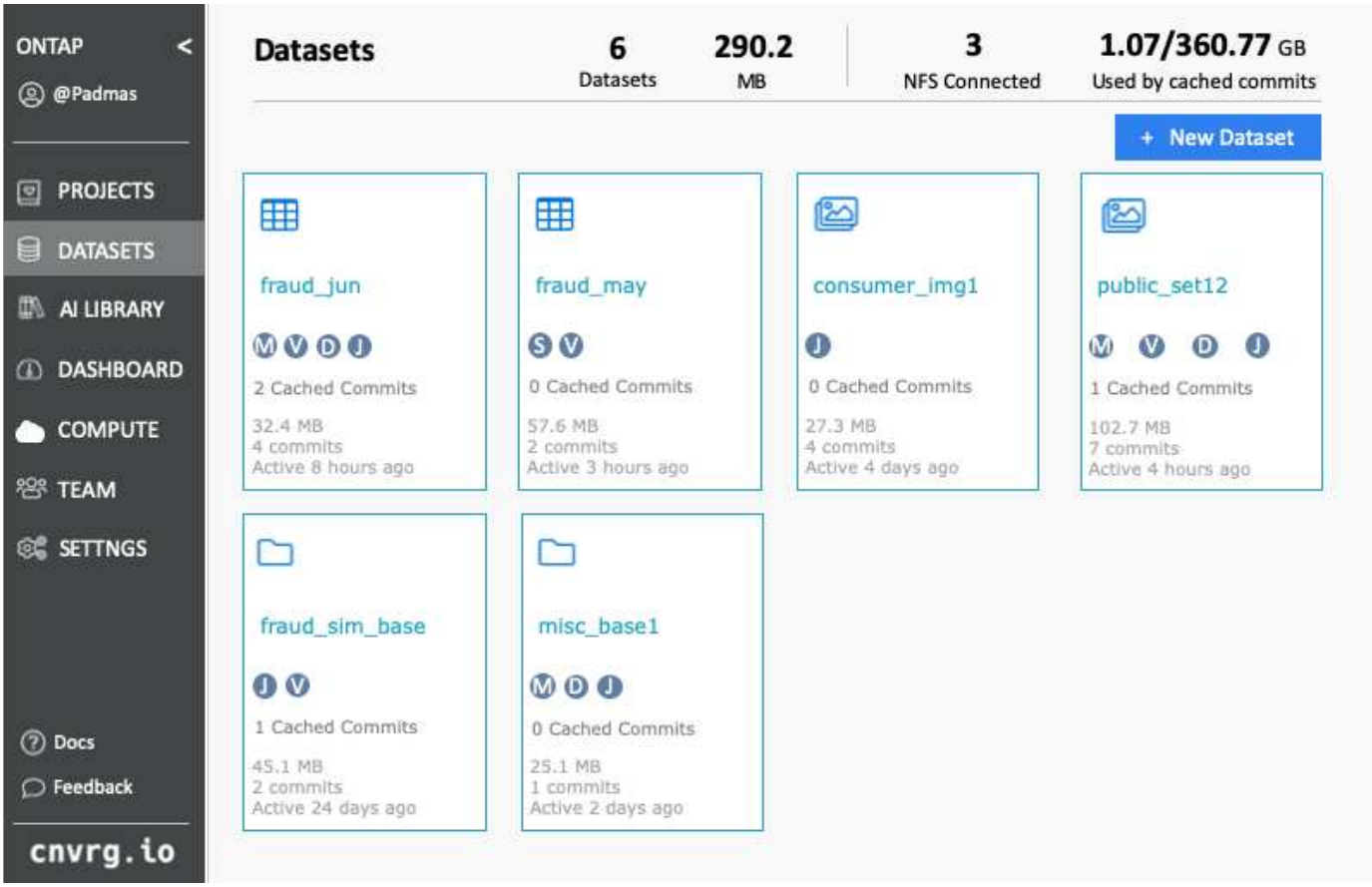
Pipeline e svantaggi convenzionali di Data Science

Una sequenza tipica di sviluppo e implementazione del modello ML prevede passaggi iterativi che includono:

- Acquisizione dei dati
- Pre-elaborazione dei dati (creazione di più versioni dei set di dati)
- Esecuzione di esperimenti multipli che coinvolgono l'ottimizzazione degli hyperparameter, modelli diversi e così via
- Implementazione
- Monitoringcnvrg.io ha sviluppato una piattaforma completa per automatizzare tutte le attività, dalla ricerca all'implementazione. Un piccolo esempio di schermate della dashboard relative alla pipeline è illustrato nella figura seguente.



È molto comune avere più set di dati in gioco da repository pubblici e dati privati. Inoltre, è probabile che ogni set di dati disponga di più versioni risultanti dalla pulizia dei set di dati o dall'ingegneria delle funzionalità. Una dashboard che fornisce un hub di set di dati e una versione hub è necessaria per garantire che i tool di collaborazione e coerenza siano disponibili per il team, come illustrato nella figura seguente.



La fase successiva della pipeline è la formazione, che richiede più istanze parallele di modelli di training, ciascuna associata a un dataset e a una determinata istanza di calcolo. L'associazione di un dataset a un certo esperimento con una determinata istanza di calcolo è una sfida perché è possibile che alcuni esperimenti vengano eseguiti da istanze GPU da Amazon Web Services (AWS), mentre altri esperimenti vengono eseguiti da istanze DGX-1 o DGX-2 on-premise. Altri esperimenti potrebbero essere eseguiti nei server CPU in GCP, mentre la posizione del set di dati non si trova in prossimità delle risorse di calcolo che eseguono il training. Una vicinanza ragionevole avrebbe una connettività completa a 10 GbE o più a bassa latenza dallo storage del dataset all'istanza di calcolo.

È pratica comune per i data scientist scaricare il set di dati nell'istanza di calcolo che esegue il training ed esegue l'esperimento. Tuttavia, questo approccio può comportare diversi problemi:

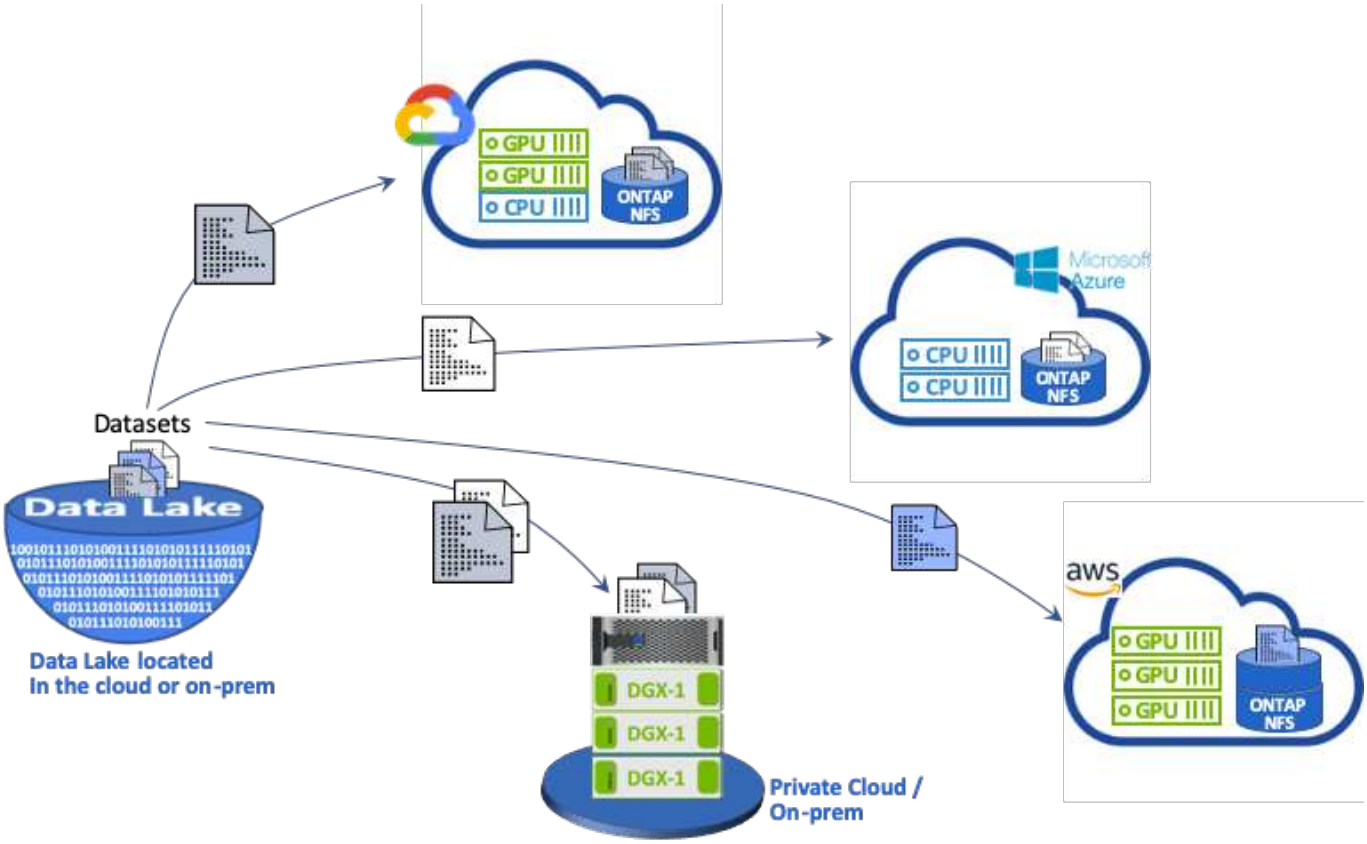
- Quando il data scientist scarica il dataset in un'istanza di calcolo, non vi sono garanzie che lo storage di calcolo integrato sia dalle performance elevate (un esempio di sistema dalle performance elevate sarebbe la soluzione NVMe ONTAP AFF A800).
- Quando il set di dati scaricato risiede in un nodo di calcolo, lo storage può diventare un collo di bottiglia quando i modelli distribuiti vengono eseguiti su più nodi (a differenza dello storage distribuito dalle performance elevate di NetApp ONTAP).
- La successiva iterazione dell'esperimento di training potrebbe essere eseguita in un'istanza di calcolo diversa a causa di conflitti di coda o priorità, creando nuovamente una distanza di rete significativa dal dataset alla posizione di calcolo.
- Gli altri membri del team che eseguono esperimenti di training sullo stesso cluster di calcolo non possono condividere questo set di dati; ciascuno esegue il (costoso) download del set di dati da una posizione arbitraria.
- Se sono necessari altri set di dati o versioni dello stesso set di dati per i successivi lavori di formazione, i data scientist devono eseguire nuovamente il (costoso) download del set di dati nell'istanza di calcolo che

esegue training.NetApp e cnvrg.io hanno creato una nuova soluzione di caching del set di dati che elimina questi ostacoli. La soluzione crea un'esecuzione accelerata della pipeline ML memorizzando nella cache i set di dati hot sul sistema storage ad alte performance ONTAP. Con ONTAP NFS, i set di dati vengono memorizzati nella cache una sola volta (e una sola volta) in un data fabric basato su NetApp (ad esempio AFF A800), che viene posizionato insieme al calcolo. Poiché lo storage NetApp ONTAP NFS ad alta velocità può servire più nodi di calcolo ML, le performance dei modelli di training sono ottimizzate, offrendo risparmi sui costi, produttività ed efficienza operativa all'organizzazione.

Architettura della soluzione

Questa soluzione di NetApp e cnvrg.io fornisce il caching dei set di dati, come mostrato nella figura seguente. Il caching dei set di dati consente agli scienziati dei dati di scegliere una versione di set di dati o set di dati desiderata e di spostarla nella cache NFS di ONTAP, che si trova in prossimità del cluster di calcolo ML. Il data scientist può ora eseguire più esperimenti senza incorrere in ritardi o download. Inoltre, tutti i tecnici che collaborano possono utilizzare lo stesso set di dati con il cluster di calcolo collegato (con la libertà di scegliere qualsiasi nodo) senza ulteriori download dal data Lake. Ai data scientist viene offerta una dashboard che tiene traccia e monitora tutti i set di dati e le versioni e fornisce una vista dei set di dati memorizzati nella cache.

La piattaforma cnvrg.io rileva automaticamente i set di dati vecchi che non sono stati utilizzati per un certo periodo di tempo e li eludono dalla cache, mantenendo spazio libero nella cache NFS per i set di dati più utilizzati. È importante notare che il caching dei set di dati con ONTAP funziona nel cloud e on-premise, fornendo così la massima flessibilità.



Concetti e componenti

In questa sezione vengono illustrati i concetti e i componenti associati al caching dei dati in un workflow ML.

Apprendimento automatico

ML sta diventando rapidamente essenziale per molte aziende e organizzazioni in tutto il mondo. Pertanto, I team IT e DevOps devono ora affrontare la sfida della standardizzazione dei carichi DI lavoro ML e del provisioning di cloud, risorse di calcolo on-premise e ibride che supportano i flussi di lavoro dinamici e intensivi richiesti dai processi E dalle pipeline ML.

Machine Learning e Kubernetes basati su container

I container sono istanze isolate dello spazio utente eseguite su un kernel del sistema operativo host condiviso. L'adozione dei container è in rapida crescita. I container offrono molti degli stessi vantaggi offerti dalle macchine virtuali (VM) per il sandboxing delle applicazioni. Tuttavia, poiché l'hypervisor e i livelli del sistema operativo guest su cui si basano le macchine virtuali sono stati eliminati, i container sono molto più leggeri.

I container consentono inoltre un efficiente packaging delle dipendenze delle applicazioni, dei tempi di esecuzione e così via direttamente con un'applicazione. Il formato di packaging dei container più comunemente utilizzato è Docker Container. Un'applicazione che è stata containerizzata nel formato Docker container può essere eseguita su qualsiasi computer in grado di eseguire i container Docker. Ciò è vero anche se le dipendenze dell'applicazione non sono presenti sul computer, perché tutte le dipendenze sono contenute nel container stesso. Per ulteriori informazioni, visitare il "Sito web di Docker".

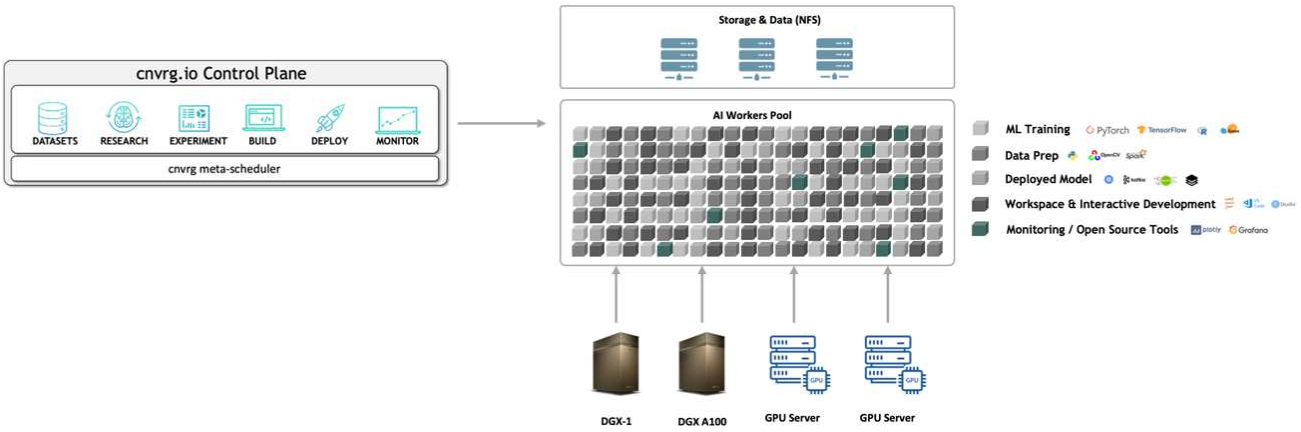
Kubernetes, il popolare container orchestrator, consente agli scienziati dei dati di lanciare processi e pipeline flessibili e basati su container. Consente inoltre ai team dell'infrastruttura di gestire e monitorare i carichi DI lavoro ML in un singolo ambiente gestito e nativo del cloud. Per ulteriori informazioni, visitare il "Sito web di Kubernetes".

cnvrg.io

Cnvrg.io è un sistema operativo ai che trasforma il modo in cui le aziende gestiscono, scalano e accelerano l'ai e lo sviluppo di data science dalla ricerca alla produzione. La piattaforma code-first è costruita dai data scientist per i data scientist e offre flessibilità per l'esecuzione on-premise o nel cloud. Grazie alla gestione dei modelli, agli MLOps e alle soluzioni DI ML continuo, cnvrg.io porta la tecnologia top di gamma ai team di data science in modo che possano dedicare meno tempo a DevOps e concentrarsi sugli algoritmi più magici. Da quando si utilizza cnvrg.io, i team di diversi settori hanno ottenuto più modelli in produzione, con un conseguente aumento del valore di business.

Meta-Scheduler cnvrg.io

cnvrg. io ha un'architettura unica che consente A IT e ingegneri di collegare diverse risorse di calcolo allo stesso piano di controllo e di fare in modo che cnvrg.io gestisca i lavori ML in tutte le risorse. Ciò significa che può collegare più cluster Kubernetes on-premise, server VM e account cloud ed eseguire carichi DI lavoro ML su tutte le risorse, come mostrato nella figura seguente.



Caching dei dati cnvrg.io

cnvrg.io consente agli scienziati dei dati di definire le versioni dei set di dati hot e cold con la sua tecnologia di caching dei dati. Per impostazione predefinita, i set di dati vengono memorizzati in un database di storage a oggetti centralizzato. Quindi, i data scientist possono memorizzare nella cache una versione specifica dei dati sulla risorsa di calcolo selezionata per risparmiare tempo durante il download e quindi aumentare LO sviluppo E la produttività DI ML. I set di dati memorizzati nella cache e non utilizzati per alcuni giorni vengono cancellati automaticamente dal NFS selezionato. Il caching e la cancellazione della cache possono essere eseguiti con un singolo clic; non sono richiesti né processi di codifica, NÉ OPERAZIONI IT o DevOps.

Flussi cnvrg.io e PIPELINE ML

Cnvrg.io Flows è uno strumento per la creazione di pipeline ML di produzione. Ogni componente di un flusso è uno script/codice in esecuzione su un calcolo selezionato con un'immagine di base del docker. Questo design consente a data scientist e ingegneri di creare una singola pipeline che può essere eseguita sia on-premise che nel cloud. cnvrg.io garantisce lo spostamento di dati, parametri e artefatti tra i diversi componenti. Inoltre, ogni flusso viene monitorato e monitorato per ottenere una data science riproducibile al 100%.

CORE cnvrg.io

Cnvrg.io CORE è una piattaforma gratuita per la community di data science per aiutare i data scientist a concentrarsi maggiormente sulla data science e meno su DevOps. L'infrastruttura flessibile DI CORE offre ai data scientist il controllo di utilizzare qualsiasi linguaggio, framework ai o ambiente di calcolo, sia on-premise che nel cloud, in modo che possano fare ciò che fanno meglio, costruire algoritmi. Il CORE cnvrg.io può essere facilmente installato con un singolo comando su qualsiasi cluster Kubernetes.

NetApp ONTAP ai

ONTAP ai è un'architettura di riferimento per data center per i carichi di lavoro ML e deep learning (DL) che utilizza i sistemi storage NetApp AFF e i sistemi NVIDIA DGX con GPU Tesla V100. ONTAP ai si basa sul protocollo file NFS standard di settore su Ethernet da 100 GB, offrendo ai clienti un'infrastruttura ML/DL dalle performance elevate che utilizza tecnologie standard per data center per ridurre l'overhead di implementazione e amministrazione. L'utilizzo di protocolli e reti standardizzati consente a ONTAP ai di integrarsi in ambienti di cloud ibrido mantenendo al contempo coerenza e semplicità operativa. Come soluzione di infrastruttura pre-validata, ONTAP ai riduce i tempi e i rischi di implementazione e riduce significativamente l'overhead amministrativo, consentendo ai clienti di ottenere un time-to-value più rapido.

NVIDIA DeepOps

DeepOps è un progetto open source di NVIDIA che, utilizzando Ansible, automatizza l'implementazione dei cluster di server GPU in base alle Best practice. DeepOps è modulare e può essere utilizzato per varie attività di implementazione. Per questo documento e per l'esercizio di convalida descritto, DeepOps viene utilizzato per implementare un cluster Kubernetes costituito da nodi di lavoro del server GPU. Per ulteriori informazioni, visitare il "Sito Web di DeepOps".

Trident di NetApp

Trident è un orchestratore di storage open source sviluppato e gestito da NetApp che semplifica notevolmente la creazione, la gestione e il consumo dello storage persistente per i carichi di lavoro Kubernetes. Trident è un'applicazione nativa di Kubernetes che viene eseguita direttamente all'interno di un cluster Kubernetes. Con Trident, gli utenti di Kubernetes (sviluppatori, data scientist, amministratori di Kubernetes e così via) possono creare, gestire e interagire con volumi di storage persistenti nel formato standard di Kubernetes che già conoscono. Allo stesso tempo, possono sfruttare le funzionalità avanzate di gestione dei dati di NetApp e un data fabric basato sulla tecnologia NetApp. Trident astratta le complessità dello storage persistente e lo rende semplice da utilizzare. Per ulteriori informazioni, visitare il "Sito web di Trident".

NetApp StorageGRID

NetApp StorageGRID è una piattaforma di storage a oggetti software-defined progettata per soddisfare queste esigenze fornendo uno storage semplice e simile al cloud a cui gli utenti possono accedere utilizzando il protocollo S3. StorageGRID è un sistema scale-out progettato per supportare più nodi nei siti connessi a Internet, indipendentemente dalla distanza. Con il motore intelligente delle policy di StorageGRID, gli utenti possono scegliere oggetti di erasure coding tra i siti per georesilienza o replica di oggetti tra siti remoti per ridurre al minimo la latenza di accesso WAN. StorageGRID offre un eccellente data Lake di storage a oggetti primario per il cloud privato in questa soluzione.

NetApp Cloud Volumes ONTAP

Il software per la gestione dei dati NetApp Cloud Volumes ONTAP offre controllo, protezione ed efficienza ai dati degli utenti con la flessibilità dei provider di cloud pubblico, tra cui AWS, Google Cloud Platform e Microsoft Azure. Cloud Volumes ONTAP è un software per la gestione dei dati nativo del cloud basato sul software di storage NetApp ONTAP, che offre agli utenti una piattaforma di storage universale di livello superiore in grado di soddisfare le loro esigenze di dati nel cloud. La disponibilità dello stesso software di storage nel cloud e on-premise offre agli utenti il valore di un data fabric senza dover formare il personale IT in nuovi metodi per la gestione dei dati.

Per i clienti interessati ai modelli di implementazione del cloud ibrido, Cloud Volumes ONTAP è in grado di fornire le stesse funzionalità e performance leader di settore nella maggior parte dei cloud pubblici per offrire un'esperienza utente coerente e perfetta in qualsiasi ambiente.

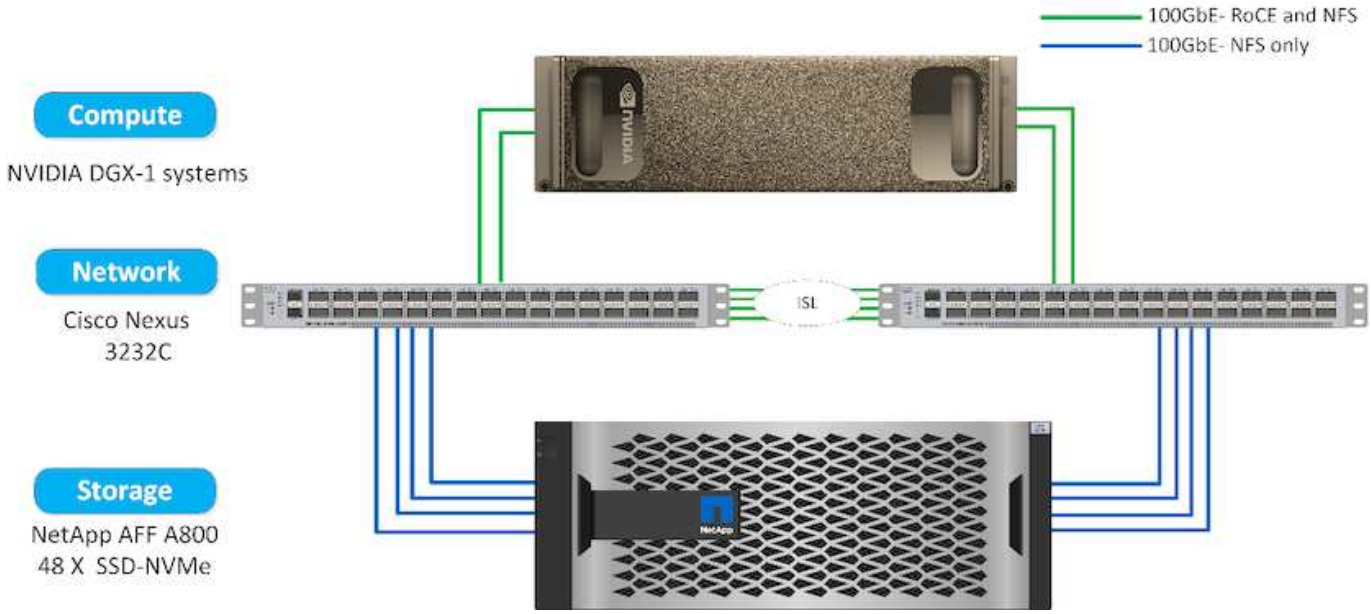
Requisiti hardware e software

In questa sezione vengono illustrati i requisiti tecnologici per la soluzione ai di ONTAP.

Requisiti hardware

Sebbene i requisiti hardware dipendano da carichi di lavoro specifici dei clienti, ONTAP ai può essere implementato su qualsiasi scala per data engineering, formazione sui modelli e deduzione di produzione da una singola GPU fino a configurazioni su scala rack per operazioni ML/DL su larga scala. Per ulteriori informazioni su ONTAP ai, vedere "Sito web ONTAP ai".

Questa soluzione è stata validata utilizzando un sistema DGX-1 per il calcolo, un sistema storage NetApp AFF A800 e Cisco Nexus 3232C per la connettività di rete. Il sistema AFF A800 utilizzato per questa convalida può supportare fino a 10 sistemi DGX-1 per la maggior parte dei carichi di lavoro ML/DL. La figura seguente mostra la topologia ONTAP ai utilizzata per il training sui modelli in questa convalida.



Per estendere questa soluzione a un cloud pubblico, Cloud Volumes ONTAP può essere implementato insieme alle risorse di calcolo della GPU del cloud e integrato in un data fabric del cloud ibrido che consente ai clienti di utilizzare le risorse appropriate per un determinato carico di lavoro.

Requisiti software

La seguente tabella mostra le versioni software specifiche utilizzate per la convalida della soluzione.

Componente	Versione
Ubuntu	18.04.4 LTS
SISTEMA OPERATIVO NVIDIA DGX	4.4.0
NVIDIA DeepOps	20.02.1
Kubernetes	1.15
Timone	3.1.0
cnvrg.io	3.0.0
NetApp ONTAP	9.6P4

Per la convalida di questa soluzione, Kubernetes è stato implementato come cluster a nodo singolo nel sistema DGX-1. Per le implementazioni su larga scala, è necessario implementare nodi master Kubernetes indipendenti per fornire un'elevata disponibilità dei servizi di gestione e riservare preziose risorse DGX per i carichi di lavoro ML e DL.

Dettagli sulla distribuzione e sulla convalida della soluzione

Le sezioni seguenti illustrano i dettagli dell'implementazione e della convalida della

soluzione.

Implementazione di ONTAP ai

L'implementazione di ONTAP ai richiede l'installazione e la configurazione dell'hardware di rete, calcolo e storage. Le istruzioni specifiche per l'implementazione dell'infrastruttura ai di ONTAP esulano dall'ambito di questo documento. Per informazioni dettagliate sull'implementazione, vedere "NVA-1121-DEPLOY: NetApp ONTAP ai, basato su NVIDIA".

Per la convalida di questa soluzione, è stato creato un singolo volume e montato sul sistema DGX-1. Tale punto di montaggio è stato quindi montato sui container per rendere i dati accessibili per la formazione. Per implementazioni su larga scala, NetApp Trident automatizza la creazione e il montaggio dei volumi per eliminare i costi amministrativi e consentire la gestione delle risorse da parte dell'utente finale.

Implementazione di Kubernetes

Per implementare e configurare il cluster Kubernetes con NVIDIA DeepOps, eseguire le seguenti operazioni da un host di distribuzione jump:

1. Scaricare NVIDIA DeepOps seguendo le istruzioni sul "Pagina introduttiva" Sul sito NVIDIA DeepOps GitHub.
2. Implementare Kubernetes nel cluster seguendo le istruzioni sul "Guida all'implementazione di Kubernetes" Sul sito NVIDIA DeepOps GitHub.

> Affinché l'implementazione di DeepOps Kubernetes funzioni, lo stesso utente deve esistere su tutti i nodi master e worker di Kubernetes.

Se l'implementazione non riesce, modificare il valore di `kubectl_localhost` a false in `deepops/config/group_vars/k8s-cluster.yml` e ripetere il punto 2. Il `Copy kubectl binary to ansible host` attività, che viene eseguita solo quando il valore di `kubectl_localhost` È vero, si basa sul modulo fetch Ansible, che presenta problemi noti di utilizzo della memoria. A volte, questi problemi di utilizzo della memoria possono causare un errore nell'attività. Se l'operazione non riesce a causa di un problema di memoria, il resto dell'operazione di implementazione non viene completata correttamente.

Se l'implementazione viene completata correttamente dopo aver modificato il valore di `kubectl_localhost` a. `false`, quindi è necessario copiare manualmente `kubectl binary` Da un nodo master Kubernetes all'host di salto per l'implementazione. È possibile trovare la posizione di `kubectl binary` su un nodo master specifico eseguendo `which kubectl` comando direttamente su quel nodo.

Implementazione di cnvrg.io

Implementare il CORE cnvrg utilizzando Helm

Helm è il modo più semplice per implementare rapidamente cnvrg utilizzando qualsiasi cluster, on-premise, Minikube o qualsiasi cluster cloud (come AKS, EKS e GKE). Questa sezione descrive come cnvrg è stato installato su un'istanza on-premise (DGX-1) con Kubernetes installato.

Prerequisiti

Prima di completare l'installazione, è necessario installare e preparare le seguenti dipendenze sul computer locale:

- Kubectl
- Timone 3.x
- Kubernetes cluster 1.15+

Implementazione con Helm

1. Per scaricare i grafici di comando più aggiornati, eseguire il seguente comando:

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

2. Prima di implementare cnvrg, è necessario disporre dell'indirizzo IP esterno del cluster e del nome del nodo su cui verrà implementato cnvrg. Per implementare cnvrg in un cluster Kubernetes on-premise, eseguire il seguente comando:

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

3. Eseguire `helm install` comando. Tutti i servizi e i sistemi vengono installati automaticamente sul cluster. Il processo può richiedere fino a 15 minuti.
4. Il `helm install` il comando può richiedere fino a 10 minuti. Una volta completata l'implementazione, accedere all'URL del cnvrg appena distribuito o aggiungere il nuovo cluster come risorsa all'interno dell'organizzazione. Il `helm` Il comando indica l'URL corretto.

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

5. Quando lo stato di tutti i container è in esecuzione o completo, cnvrg è stato implementato correttamente. Dovrebbe essere simile al seguente output di esempio:

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

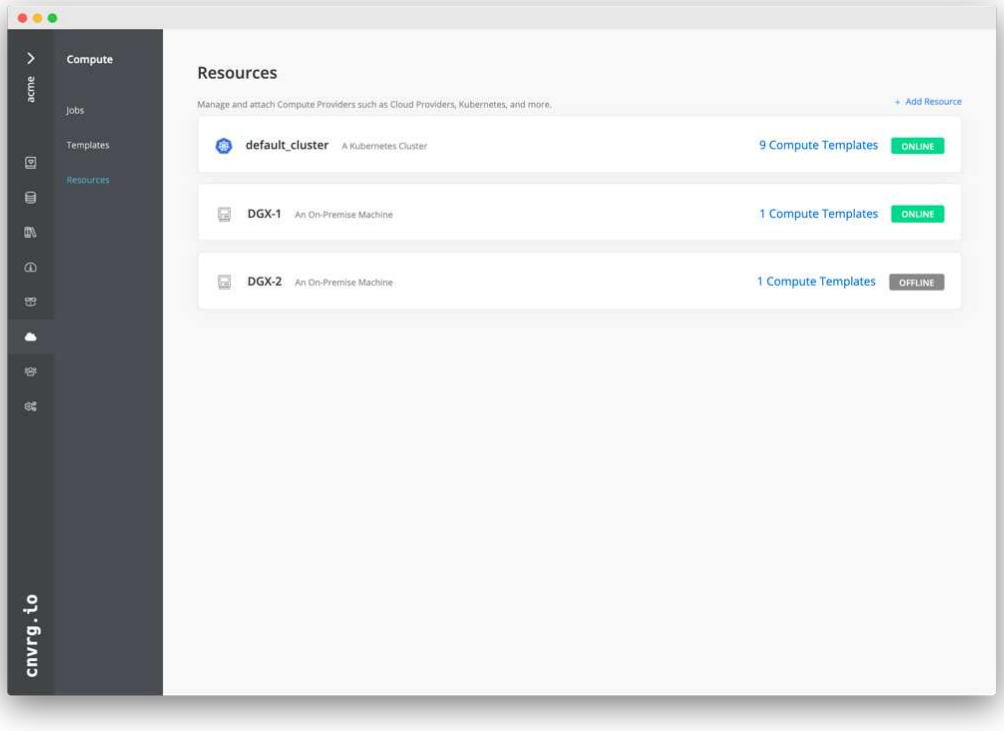
Formazione sul modello di visione artificiale con ResNet50 e il set di dati radiologici Chest

Il sistema operativo ai cnvrg.io è stato implementato su una configurazione Kubernetes su un'architettura NetApp ONTAP ai basata sul sistema NVIDIA DGX. Per la convalida, abbiamo utilizzato il set di dati radiologici NIH Chest costituito da immagini anonimizzate dei raggi X del torace. Le immagini erano in formato PNG. I dati sono stati forniti dal NIH Clinical Center e sono disponibili tramite "Sito di download NIH". Abbiamo utilizzato un campione di 250 GB dei dati con 627, 615 immagini in 15 classi.

Il set di dati è stato caricato sulla piattaforma cnvrg ed è stato memorizzato nella cache di un'esportazione NFS dal sistema di storage NetApp AFF A800.

Impostare le risorse di calcolo

L'architettura cnvrg e la funzionalità di meta-scheduling consentono a tecnici e professionisti IT di collegare diverse risorse di calcolo a una singola piattaforma. Nella nostra configurazione, abbiamo utilizzato lo stesso cluster cnvrg implementato per l'esecuzione dei carichi di lavoro di deep-learning. Se è necessario collegare altri cluster, utilizzare la GUI, come mostrato nella seguente schermata.



Caricare i dati

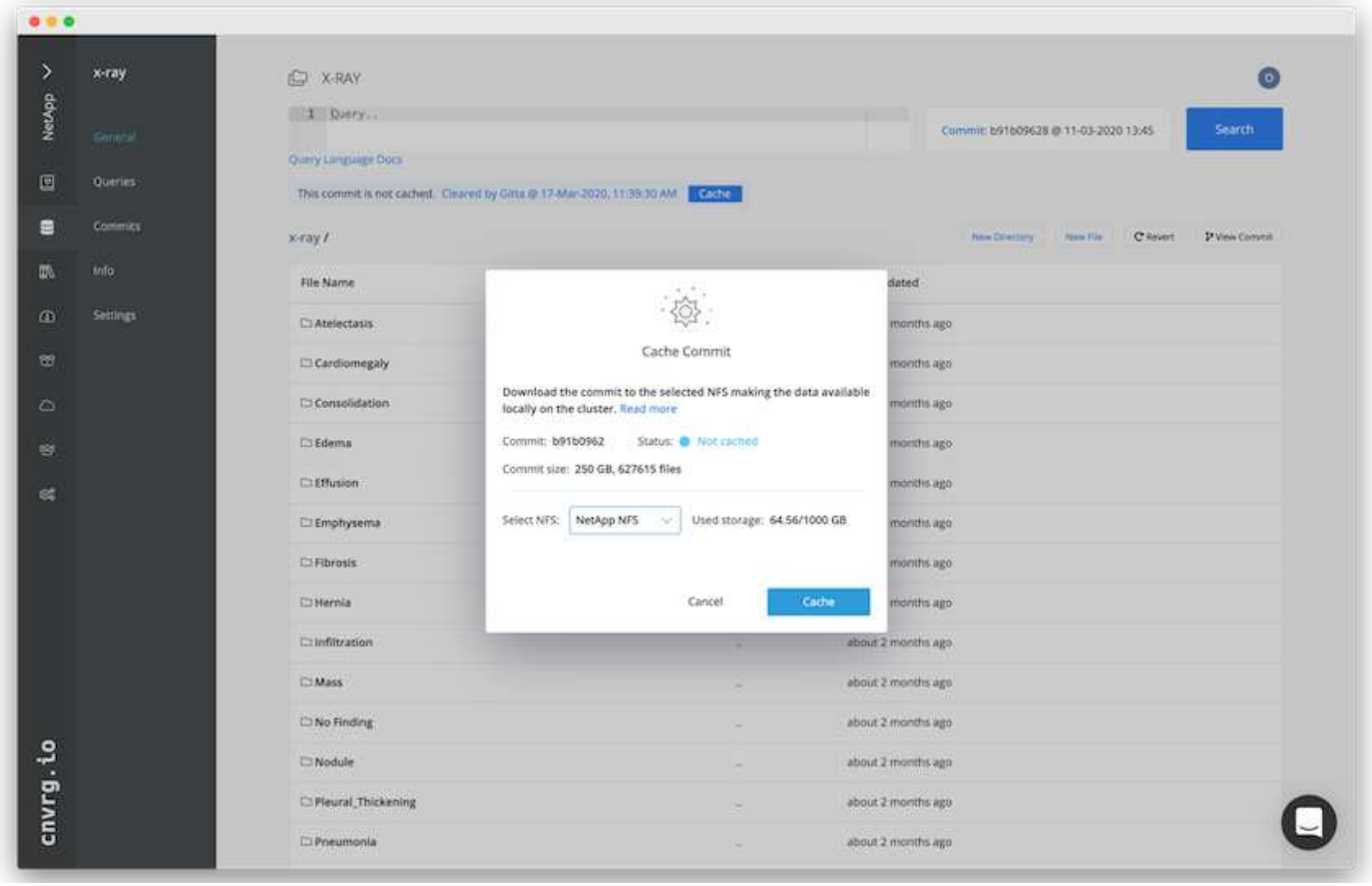
Per caricare i dati sulla piattaforma cnvrg, è possibile utilizzare la GUI o la CLI cnvrg. Per i set di dati di grandi dimensioni, NetApp consiglia di utilizzare CLI perché si tratta di uno strumento potente, scalabile e affidabile in grado di gestire un gran numero di file.

Per caricare i dati, attenersi alla seguente procedura:

1. Scaricare il "CLI cnvrg".
2. accedere alla directory dei raggi x.
3. Inizializzare il set di dati nella piattaforma con `cnvrg data init` comando.
4. Caricare tutti i contenuti della directory nel data Lake centrale con `cnvrg data sync` Command.una volta caricati i dati nell'archivio centrale di oggetti (StorageGRID, S3 o altri), è possibile navigare con la GUI. La figura seguente mostra un file PNG di immagine della fibrosi a raggi X del torace caricato. Inoltre, cnvrg consente di eseguire la versione dei dati in modo che qualsiasi modello creato possa essere riprodotto fino alla versione dei dati.

Dati di cach

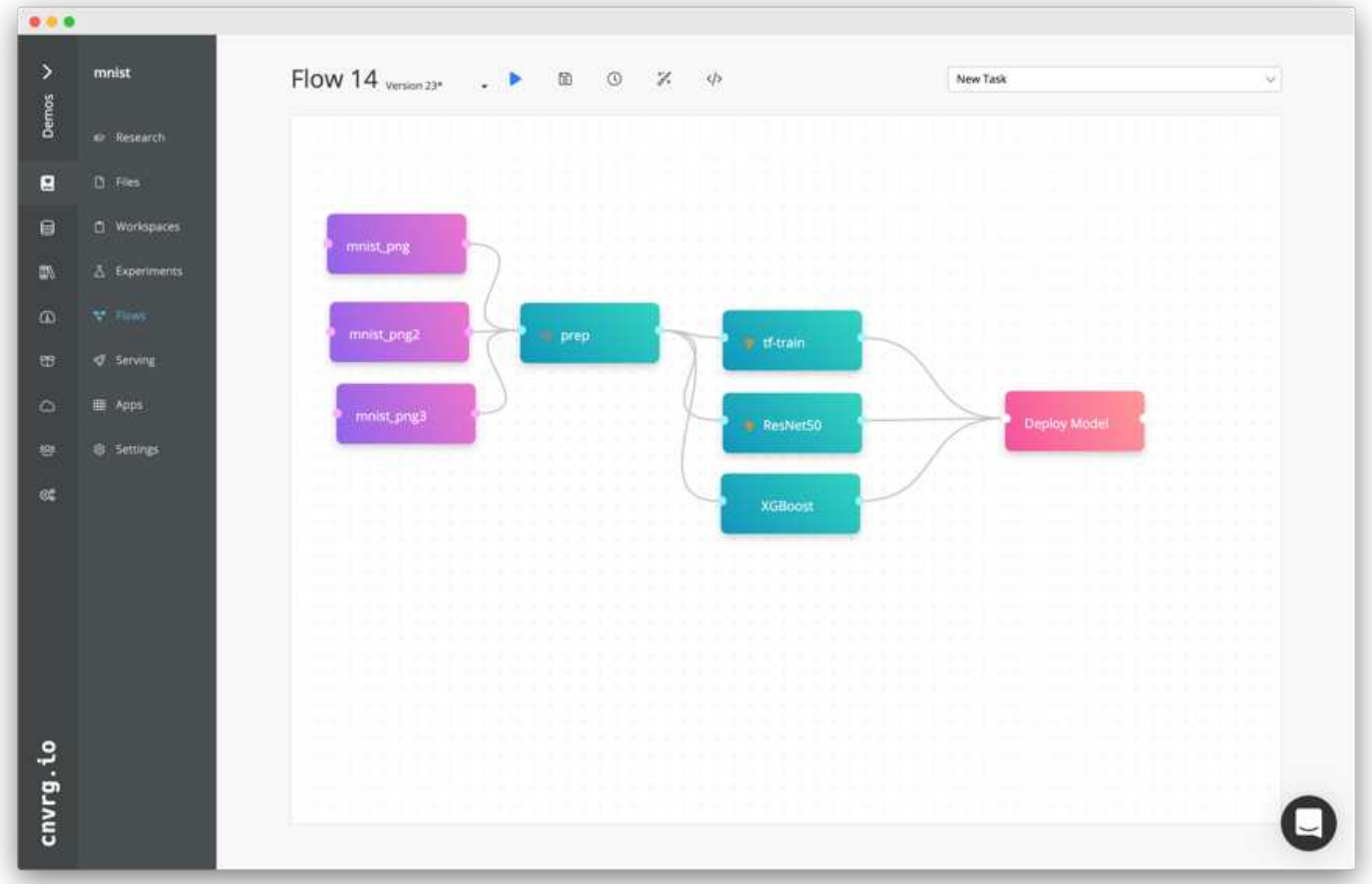
Per accelerare il training ed evitare il download di oltre 600.000 file per ciascun modello di training ed esperimento, abbiamo utilizzato la funzionalità di caching dei dati dopo che i dati sono stati inizialmente caricati nell'archivio centrale di oggetti data-Lake.



Dopo che gli utenti hanno fatto clic su cache, cnvrg scarica i dati nel relativo commit specifico dall'archivio remoto di oggetti e li memorizza nella cache del volume NFS di ONTAP. Al termine, i dati saranno disponibili per il training istantaneo. Inoltre, se i dati non vengono utilizzati per alcuni giorni (ad esempio, per la formazione o l'esplorazione del modello), cnvrg cancella automaticamente la cache.

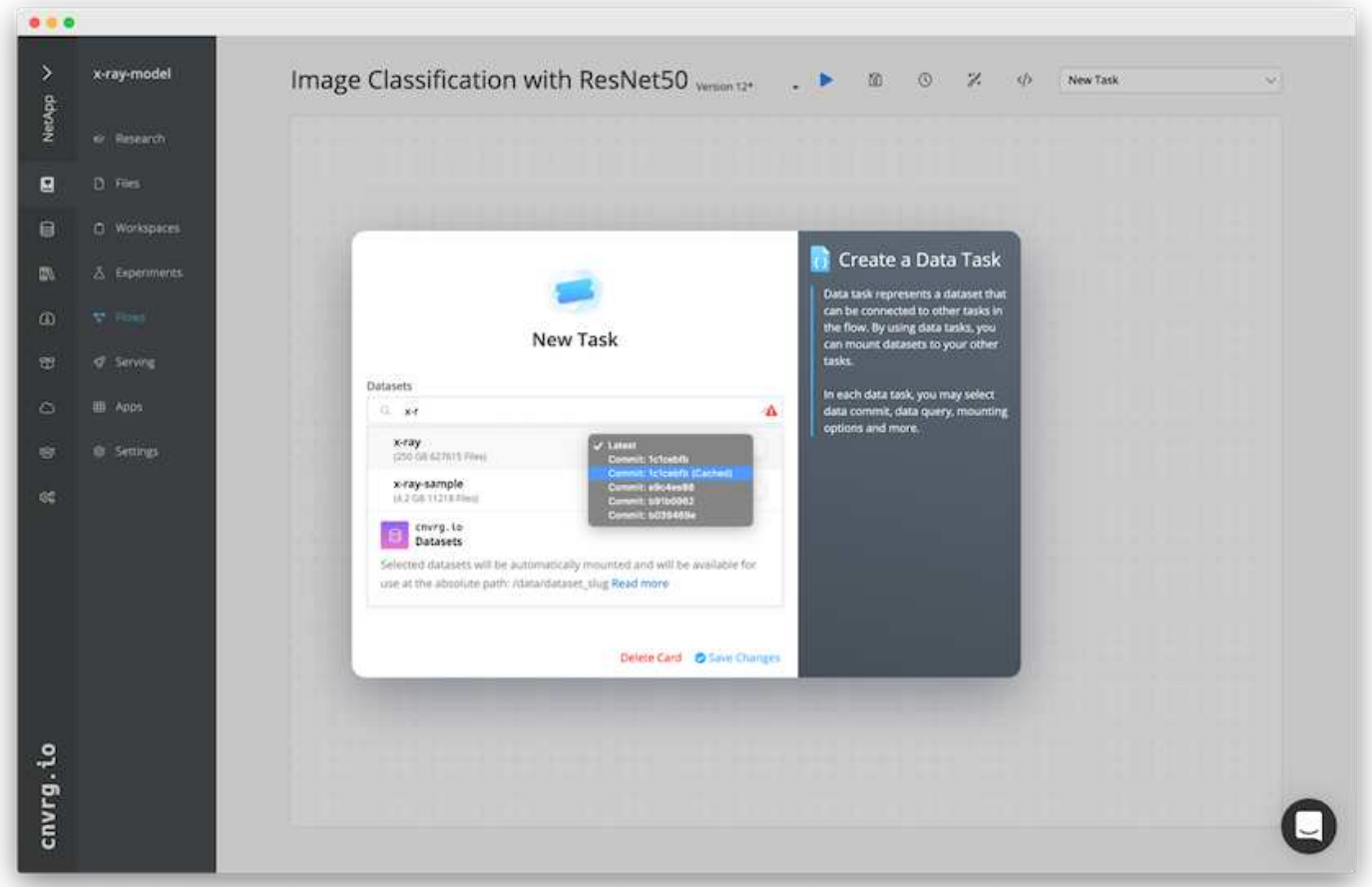
Crea una pipeline ML con i dati memorizzati nella cache

Cnvrg Flows consente di creare facilmente pipeline ML di produzione. I flussi sono flessibili, possono funzionare per qualsiasi tipo di caso d'utilizzo DI ML e possono essere creati attraverso la GUI o il codice. Ogni componente di un flusso può essere eseguito su una diversa risorsa di calcolo con un'immagine Docker diversa, il che rende possibile la creazione di cloud ibrido e pipeline ML ottimizzate.



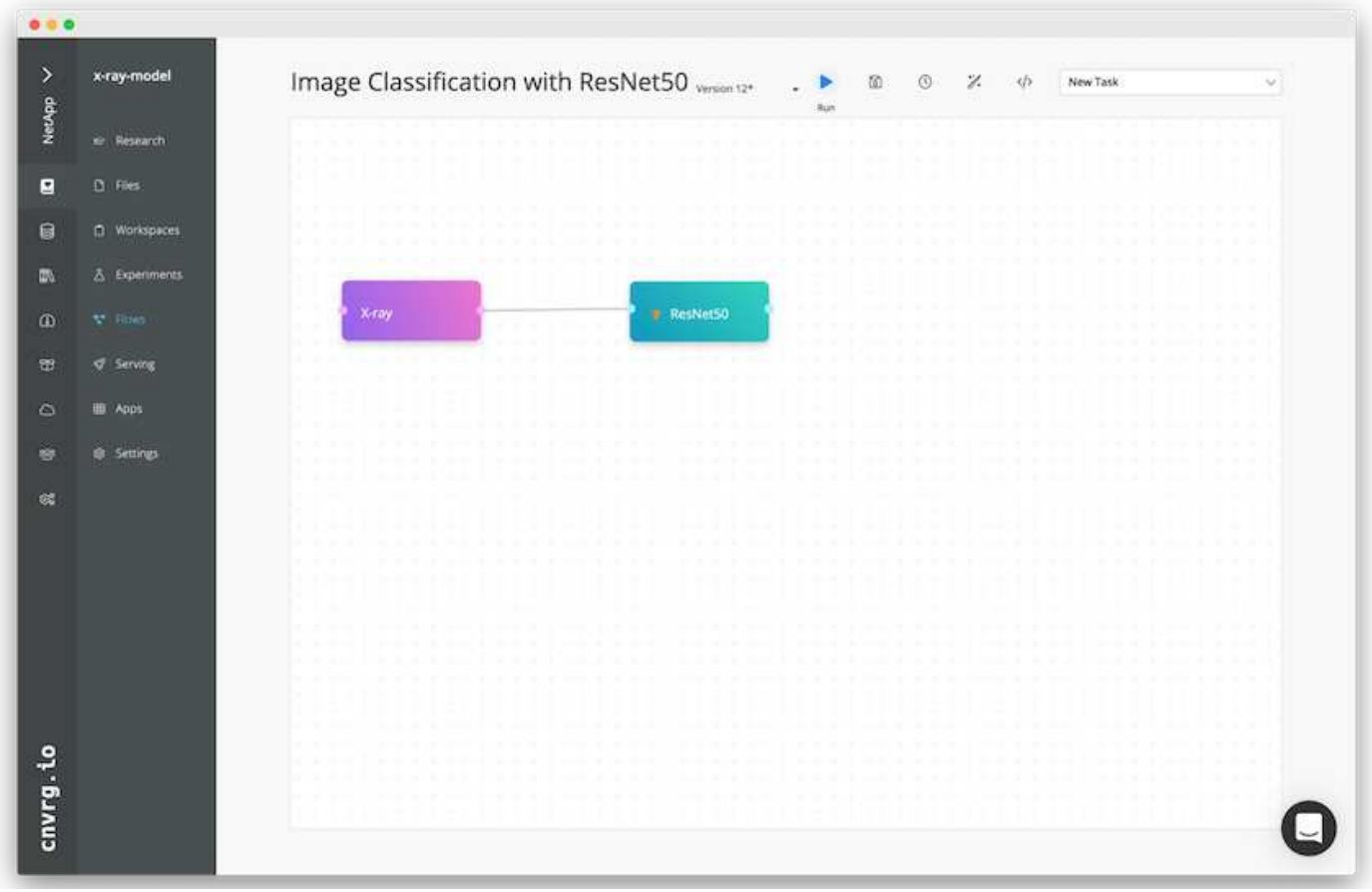
Creazione del flusso di raggi X del torace: Impostazione dei dati

Abbiamo aggiunto il nostro set di dati a un flusso appena creato. Quando si aggiunge il dataset, è possibile selezionare la versione specifica (commit) e indicare se si desidera la versione memorizzata nella cache. In questo esempio, è stato selezionato il commit memorizzato nella cache.



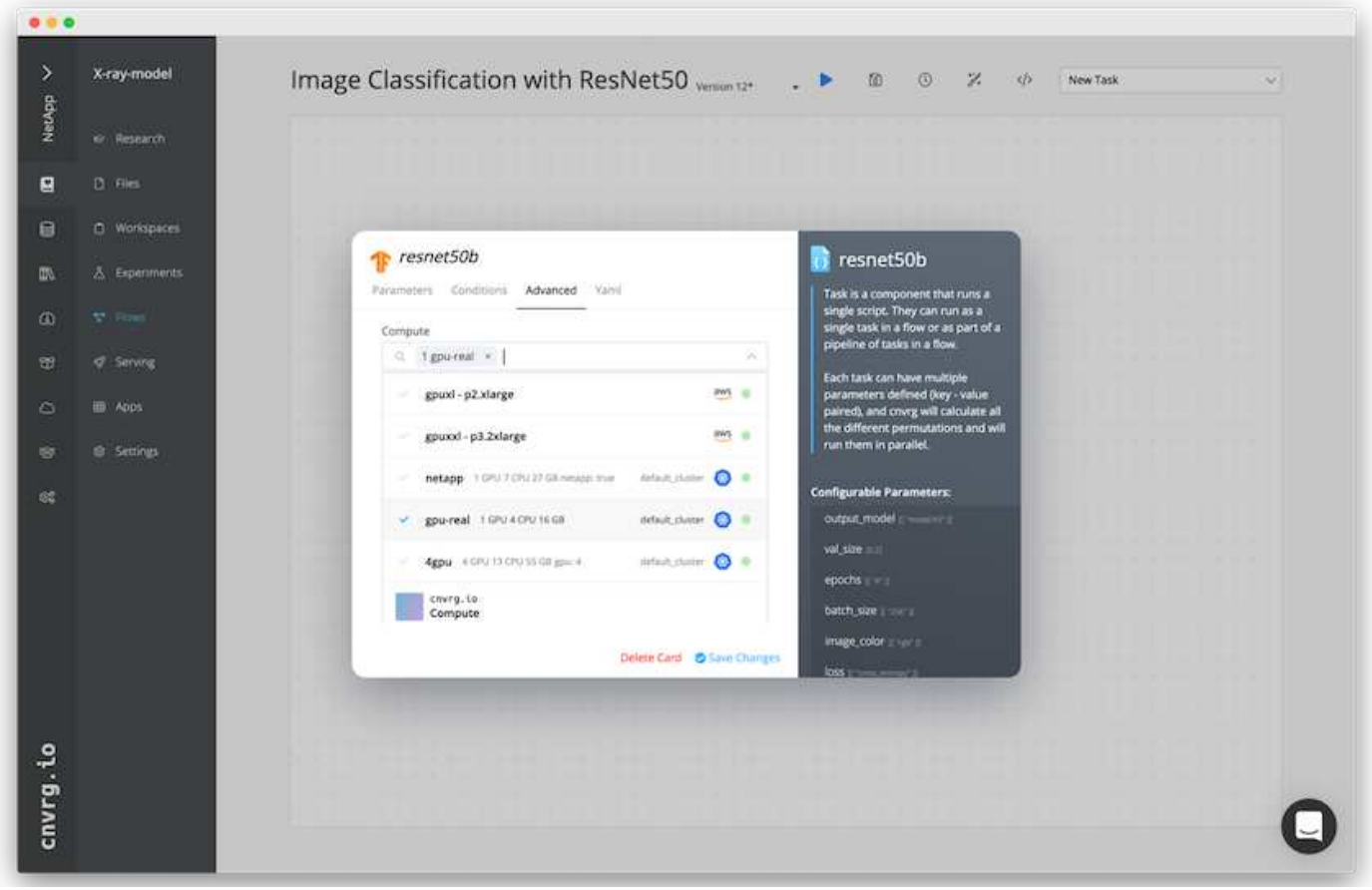
Creazione del flusso di raggi X del torace: Impostazione del modello di training: ResNet50

Nella pipeline, è possibile aggiungere qualsiasi tipo di codice personalizzato desiderato. In cnvrg è disponibile anche la libreria ai, una raccolta di componenti ML riutilizzabili. Nella libreria ai sono presenti algoritmi, script, origini dati e altre soluzioni che possono essere utilizzate in qualsiasi ML o flusso di deep learning. In questo esempio, è stato selezionato il modulo ResNet50 preinstallato. Abbiamo utilizzato parametri predefiniti come batch_size:128, epochs:10 e molto altro ancora. Questi parametri possono essere visualizzati nei documenti della ai Library. La seguente schermata mostra il nuovo flusso con il set di dati radiologici collegato a ResNet50.



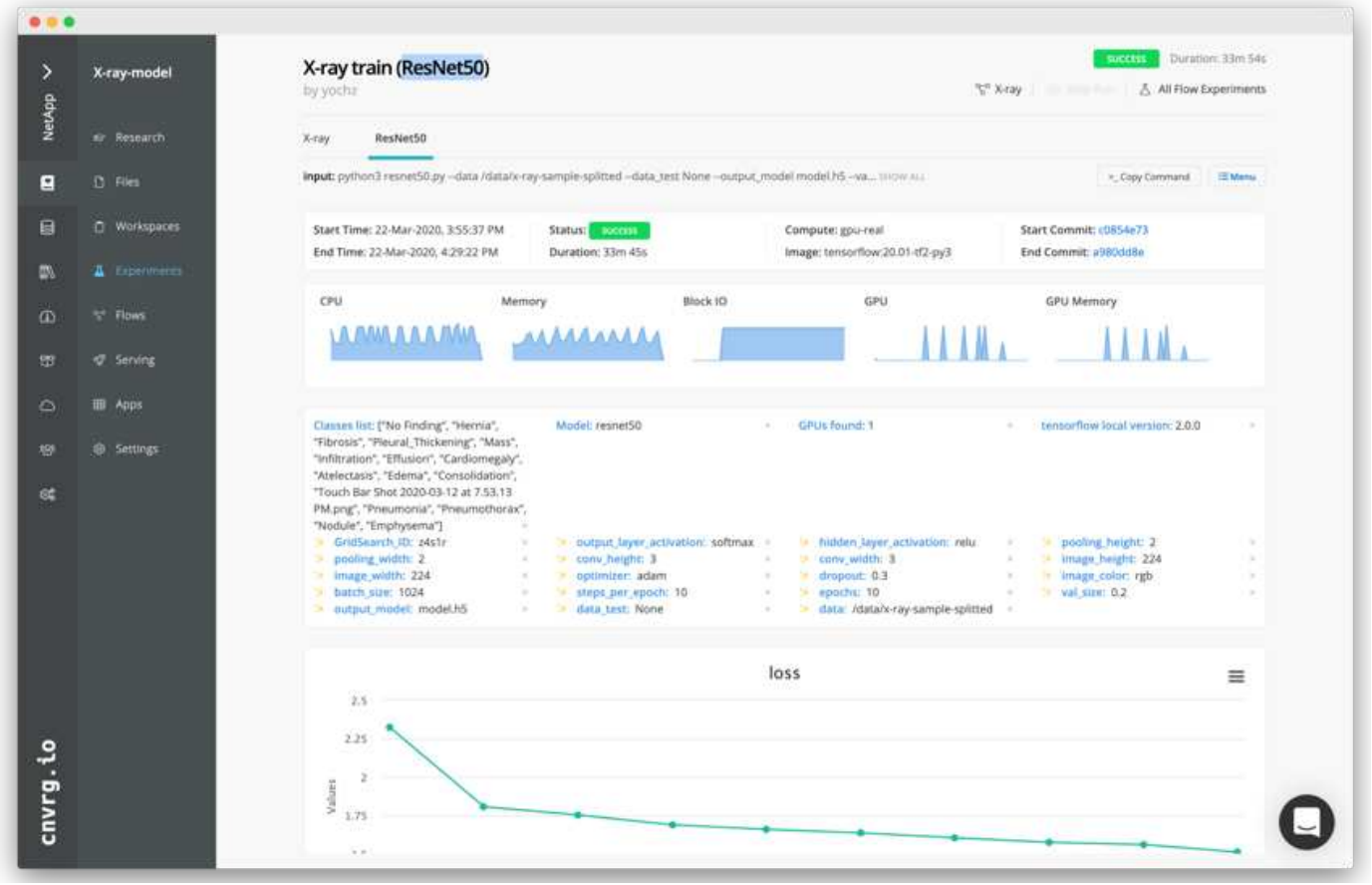
Definire la risorsa di calcolo per ResNet50

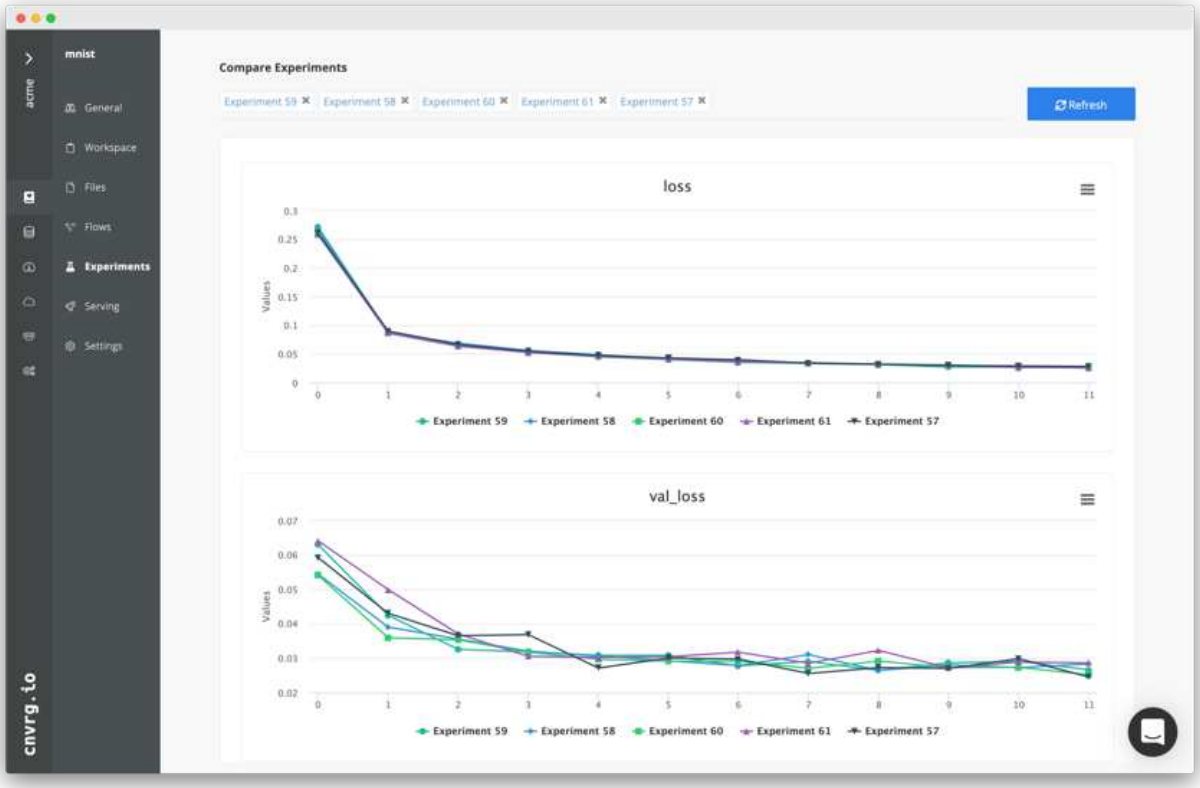
Ogni algoritmo o componente nei flussi cnvrg può essere eseguito su un'istanza di calcolo diversa, con un'immagine Docker diversa. Nella nostra configurazione, volevamo eseguire l'algoritmo di training sui sistemi NVIDIA DGX con l'architettura NetApp ONTAP ai. Nella figura seguente, è stato selezionato `gpu-real`, che è un modello di calcolo e una specifica per il nostro cluster on-premise. Abbiamo anche creato una coda di modelli e selezionato più modelli. In questo modo, se il `gpu-real` non è possibile allocare le risorse (se, ad esempio, altri data scientist le stanno utilizzando), quindi è possibile attivare la diffusione automatica del cloud aggiungendo un modello di cloud provider. La seguente schermata mostra l'utilizzo di gpu-real come nodo di calcolo per ResNet50.



Monitoraggio e monitoraggio dei risultati

Una volta eseguito un flusso, cnvrg attiva il motore di monitoraggio e tracciamento. Ogni esecuzione di un flusso viene documentata e aggiornata automaticamente in tempo reale. Hyperparameters, metriche, utilizzo delle risorse (utilizzo della GPU e altro ancora), versione del codice, artefatti, log, E così via sono disponibili automaticamente nella sezione Experiments (esperimenti), come mostrato nelle due schermate seguenti.


X-ray-model
X-ray train (ResNet50)
by yochz
Research
Files
Workspaces
Experiments
Flows
Serving
Apps
Settings
cnvrg.io
NetApp
X-ray
ResNet50
CPU
Memory
Block IO
GPU
GPU Memory
loss
Values


mnist
Compare Experiments
Refresh
General
Workspace
Files
Flows
Experiments
Serving
Settings
cnvrg.io
acme
loss
val_loss
Values
Experiment 59
Experiment 58
Experiment 60
Experiment 61
Experiment 57

Conclusione

NetApp e cnvrg.io hanno collaborato per offrire ai clienti una soluzione completa per la gestione dei dati per lo sviluppo di software ML e DL. ONTAP ai offre calcolo e storage dalle performance elevate per qualsiasi scala operativa, mentre il software cnvrg.io ottimizza i flussi di lavoro di data science e migliora l'utilizzo delle risorse.

Ringraziamenti

- Mike Oglesby, Technical Marketing Engineer, NetApp
- Santosh Rao, Senior Technical Director di NetApp

Dove trovare ulteriori informazioni

Per ulteriori informazioni sulle informazioni descritte in questo documento, consultare le seguenti risorse:

- Cnvrg.io ("https://cnvrg.io"):
  - CORE Cnvrg (piattaforma ML gratuita)

    https://cnvrg.io/platform/core
  - Documenti Cnvrg

    "https://app.cnvrg.io/docs"
- Server NVIDIA DGX-1:
  - Server NVIDIA DGX-1

    https://www.nvidia.com/en-us/data-center/dgx-1/
  - NVIDIA Tesla V100 Tensor Core GPU

    https://www.nvidia.com/en-us/data-center/tesla-v100/
  - NVIDIA GPU CLOUD (NGC)

    https://www.nvidia.com/en-us/gpu-cloud/
- Sistemi NetApp AFF:
  - Scheda informativa su AFF

    https://www.netapp.com/us/media/d-3582.pdf
  - NetApp FlashAdvantage per AFF

    https://www.netapp.com/us/media/ds-3733.pdf
  - Documentazione di ONTAP 9.x.

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

- Report tecnico di NetApp FlexGroup

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

- Storage persistente NetApp per container:
  - Trident di NetApp

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

- Matrice di interoperabilità NetApp:
  - Tool di matrice di interoperabilità NetApp

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

- Networking ai ONTAP:
  - Switch Cisco Nexus 3232C

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

  - Switch Mellanox Spectrum serie 2000

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

- Framework E tool ML:
  - DALI

    https://github.com/NVIDIA/DALI

  - TensorFlow: Un framework di apprendimento automatico open-source per tutti

    https://www.tensorflow.org/

  - Horovod: Framework di deep learning distribuito open-source di Uber per TensorFlow

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

  - Abilitazione delle GPU nell'ecosistema di runtime container

    https://devblogs.nvidia.com/gpu-containers-runtime/

  - Docker

    https://docs.docker.com

  - Kubernetes

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

  - NVIDIA DeepOps

    https://github.com/NVIDIA/deepops

  - Kubeflow

http://www.kubeflow.org/

- Jupyter notebook Server

  http://www.jupyter.org/

- Set di dati e benchmark:
  - Set di dati per radiografia toracica NIH

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

  - Xiaosong Wang, Yifan Peng, le Lu, Zhiyong Lu, Mohammadhadi Bagheri, Ronald Summers, ChestX-ray8: Database dei raggi X del torace su scala ospedaliera e benchmark sulla classificazione e localizzazione con supervisione debole delle malattie toraciche comuni, IEEE CVPR, pp 3462-3471, 2017TR-4841-0620