Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Centro Nazionale di Ricerca in HPC, Big Data e Quantum Computing

*Lucio Anderlini, **Giulio Bianchini,** Diego Ciangottini, Federica Fanzago, Rosa Petrini, Massimo Sgaravatto, Daniele Spiga, Tommaso Tedeschi, Antonino Troja*

Prime esperienze nell'integrazione di risorse eterogenee con InterLink: stato e sviluppi futuri

Workshop sul Calcolo nell'INFN, Biodola (LI), 27/05/2025

1 Introduzione

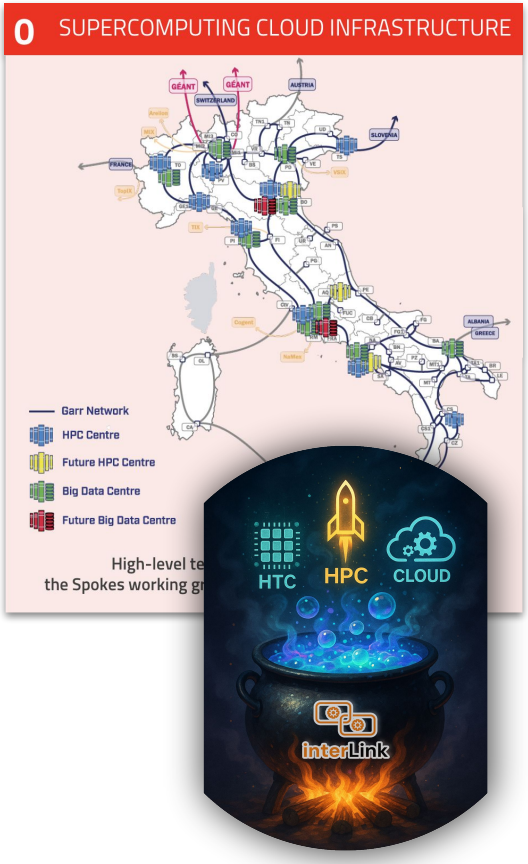
- *Integrazione di risorse eterogenee (HPC, HTC, Cloud)*
- *Provisioning trasparente per l'utente*
- *Gestione backend eterogenei con interLink*

2 Esperienze d'uso

- ***Proof of Concept ICSC per integrazione CINECA - Testbed DataCloud WP6***
- ***Piattaforma AI INFN***
- ***Integrazione con Infrastrutture HPC - Vega, Julich***
- ***High Rate Analysis Platform***

3 Sviluppi futuri

Introduzione



HPC, HTC and Cloud

Integrare risorse eterogenee come High-Performance Computing (**HPC**), High-Throughput Computing (**HTC**) e **Cloud computing** da provider distribuiti è una sfida complessa e attuale.

*Come è possibile garantire un **uso efficiente delle risorse**, una **gestione uniforme dei workload** e la **trasparenza** dell'eterogeneità verso l'utente finale, integrando risorse che hanno modelli di provisioning differenti?*

interLink

consente l'esecuzione di qualsiasi container gestito da Kubernetes su backend eterogenei,
riducendo al minimo i requisiti per l'utente e per il provider.

https://interlink-hq.github.io/interLink/

interLink in a nutshell

✅ **Cosa**

- **Gestione unificata di backend e modelli di provisioning diversi**
  - un unico set di API per utilizzare risorse eterogenee (Slurm, HTCondor, Kubernetes...) senza esporre la complessità all'utente finale.

✅ **Come**

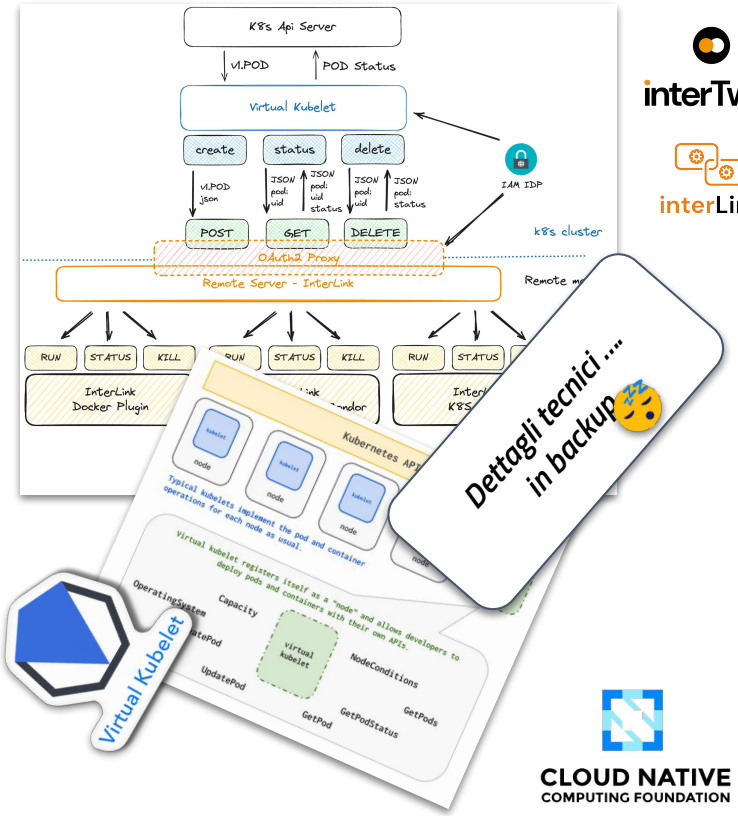
- **Esecuzione remota di POD su qualsiasi backend**
  - Estensione di Virtual Kubelet tramite un API layer generico per delegare l'esecuzione di workload su infrastrutture diverse.

✅ **Tecnologie abilitanti**

- **VK Core**: simula un nodo e riceve richieste da Kubernetes.
- **InterLink**: API server stateless, media le richieste tra VK e Plugin.
- **Plugin**: esegue i container sull'infrastruttura e restituisce i risultati.

✅ **Alcuni highlights:**

- Capace di gestire risorse eterogenee (GPU/CPU - FPGA)
- Architettura modulare, personalizzabile tramite plugin.
- Sui target provider nessuna dipendenza imposta da Kubernetes: estensione trasparente.



interTwin

interLink

CLOUD NATIVE COMPUTING FOUNDATION

https://www.cncf.io/projects/interlink/

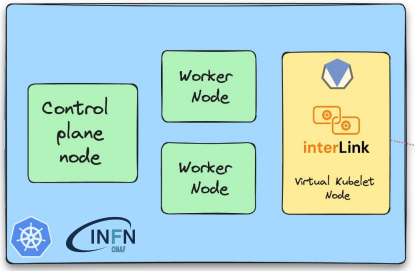
Esperienze d’uso

Disclaimer

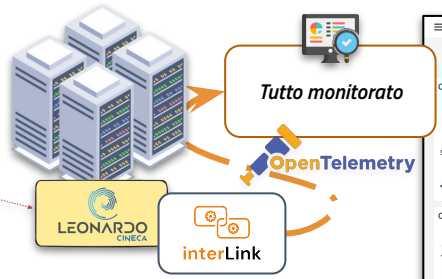
Attualmente sono in corso molteplici iniziative dove si sta testando l'integrazione di interLink.
In questa presentazione verranno discussi solo alcuni. Altri sono parte di interventi al WorkShop.

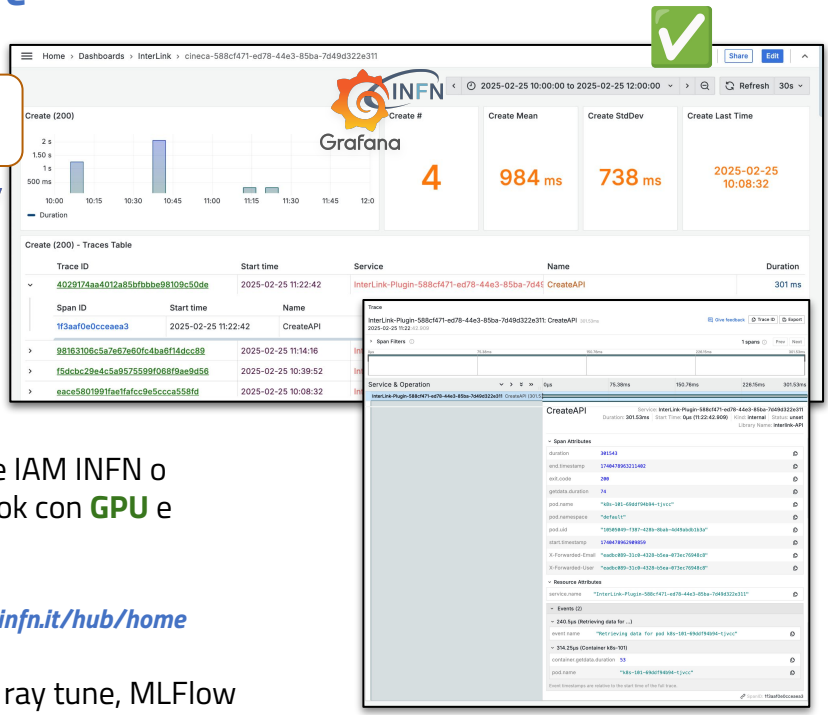
- *Advanced Tracking Analysis in Space Experiments with Graph Neural Networks* **F. Cuna**
- *INFN Cloud Kubeflow as a Platform (KaaP) and a ChatBot use case* **M. Gattari**
- *ICSC HaMMoN: Intelligenza Artificiale e Cloud Computing per il Monitoraggio dei Rischi Climatici* **A. Casale**
- ***Studi su mpi (CNAF)***
- ***interTwin [https://github.com/interTwin-eu/itwinai]***

PoC ICSC/TeRABIT per implementazione DataLake Nazionale INFN-CINECA HPC



Offload





Ambiente di test WP6 progettato per sfruttare il meccanismo di offloading tramite InterLink, al fine di accedere alle risorse HPC di LEONARDO/Cineca (**beta users sono BENVENUTI !**).

Da semplici POD …
Fino a JHUB (con autenticazione IAM INFN o PoC-ICSC) per spawn di notebook con **GPU** e **Desktop remoto.**

https://jhub.131.154.99.68.myip.cloud.infn.it/hub/home

Qualsiasi altra integrazione (i.e. ray tune, MLFlow etc. supportabile come R/D)

Confluence ***https://confluence.infn.it/x/BgAwK***

https://monitoring.cloud.infn.it:3000/goto/bKSucbxNR?orgId=1

PIATTAFORMA AI_INFN

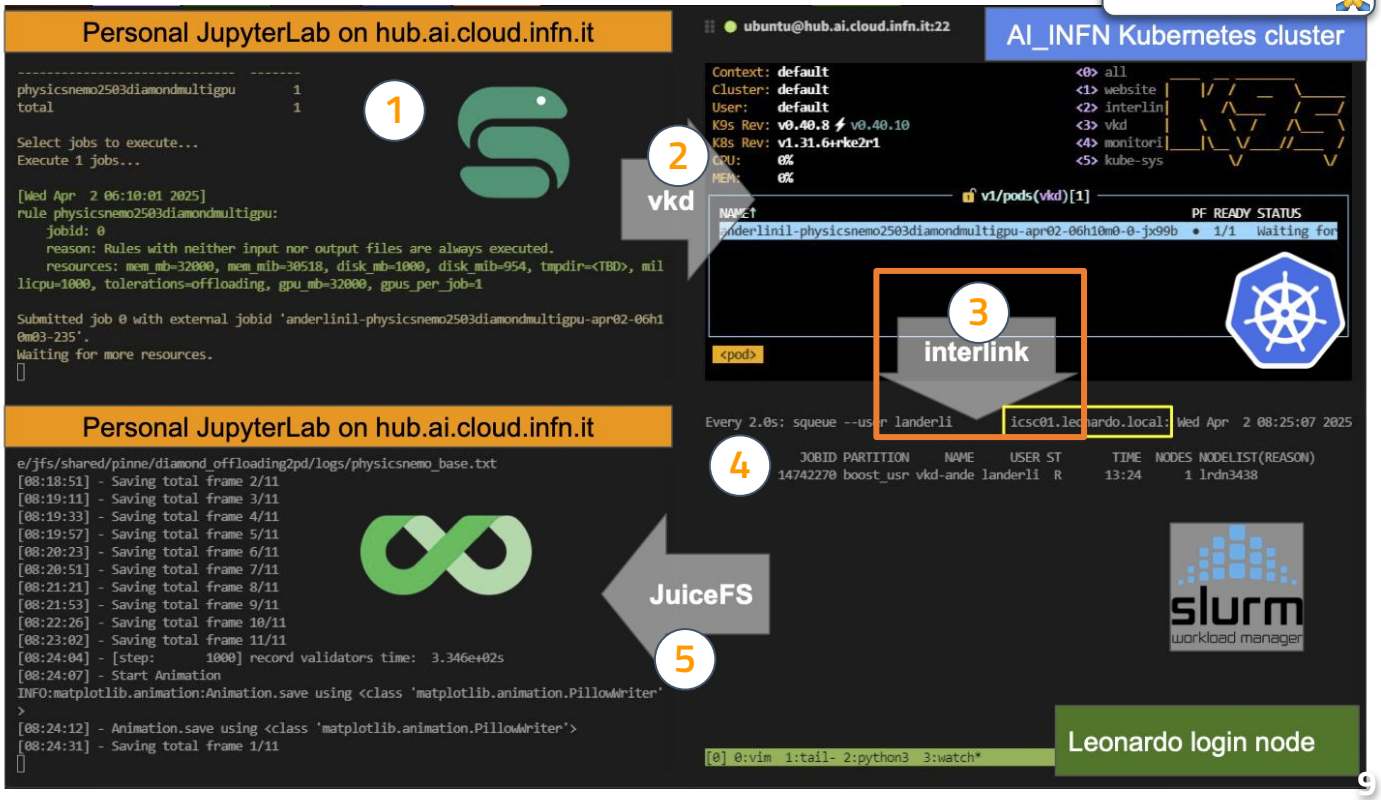
AI_INFN Artificial Intelligence technologies for INFN research

credits L. Anderlini

In fase di test (anche) con HPC bubble di Padova

use case selezionato anche per il PoC ICSC/TeraBit

1. Tramite la piattaforma AI INFN (dettagli -> talk L. Anderlini & talk R. Petrini)) l'utente avvia un JupyterLab dal quale **sottomette un job Modulus via Snakemake**
2. La sottomissione del Job avviene tramite il **vk-dispatcher**
3. Il Job viene schedulato ad un nodo virtuale **InterLink** che lo inoltrerà' verso la risorsa HPC esterna
4. Il job inoltrato al sistema HPC (LEONARDO) e' sottomesso allo SLURM workload manager tramite il **plugin SLURM**
5. Alcuni metadati vengono scritti su **JuiceFS** ed accessibili all'utente dal suo JupyterLab notebook



Analisi interattiva su risorse distribuite (grid, bubble, hpc...)

✅ **Sinergia con use case Spoke2/3**

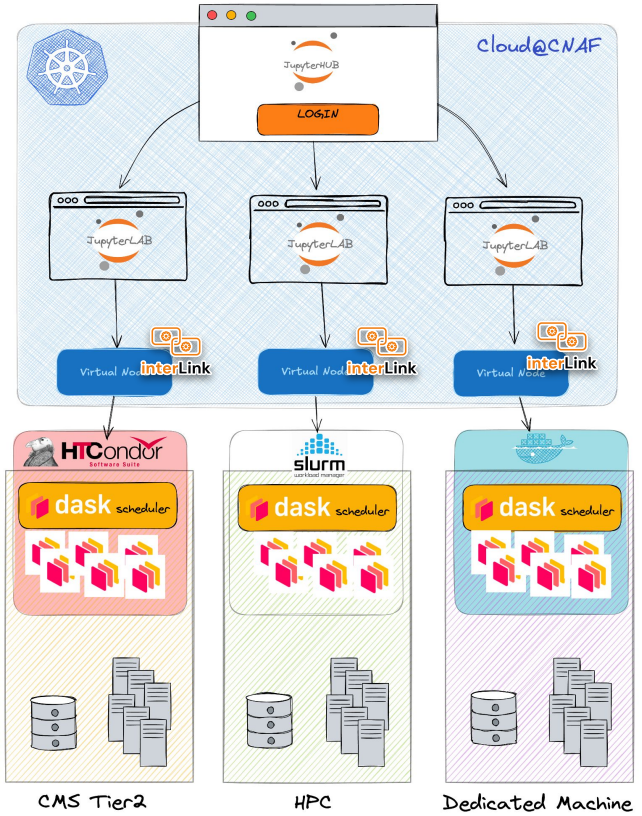
- High rate analysis
- Scientific hub

✅ **Scale out per "Analysis Facility" di CMS**

- Integrazione delle risorse HTC Tier2 CMS-IT
- Integrazione con bubble HPC (in modo opportunistico) in collaborazione con infn-pd
  - studi in vista di **High-Luminosity Large Hadron Collider**

✅ **Attività in sinergia con il CERN ,** in valutazione per

- Analysis facility a CERN (Swan)
- Sistema di provisioning GPU



Highlights from the EU context (VEGA - Slovenian EuroHPC)

Integrazione con framework ML (sinergia progetto EU interTwin)

interTwin

Caso d'uso basato su un algoritmo **GAN 3D** sviluppato al CERN che e' stato integrato con mlflow per il monitoraggio e la gestione degli esperimenti ML

Le **GPUs** utilizzate messe a disposizione dai sistemi HPC utilizzati lungo l'intera chain grazie ad InterLink

Le **annotazioni** sui POD sono un aspetto chiave e uno strumento molto potente: tramite queste e' possibile inviare le richieste specifiche al sistema remoto

In tutto questo, **dove si trova il Virtual Kubelet**? Il VK fa parte di un cluster k8s fornito dalle risorse cloud dell'INFN

Flusso di lavoro ML che può essere eseguito anche a Cineca

Dal punto di vista dei provider di risorse...

https://github.com/interlink-hq

- Il modello a **plugin** offre l'opportunità di integrare diversi sistemi di gestione di risorse di calcolo.
- Non tutti i plugin attualmente sono allo stesso livello di maturità'
- esistono plugin personalizzati da utenti finale (punto di forza)

Soluzioni attuali
cosa si può integrare oggi

slurm-plugin
https://github.com/interlink-hq/interlink-slurm-plugin

HTCondor-plugin
https://github.com/interlink-hq/interlink-htcondor-plugin

kubernetes-plugin
https://baltig.infn.it/atroja/k8sidecar
https://github.com/interlink-hq/interlink-kubernetes-plugin
https://baltig.infn.it/mgattari/interlink-kubernetes-plugin

kueue-plugin
https://github.com/interlink-hq/interlink-kueue-plugin

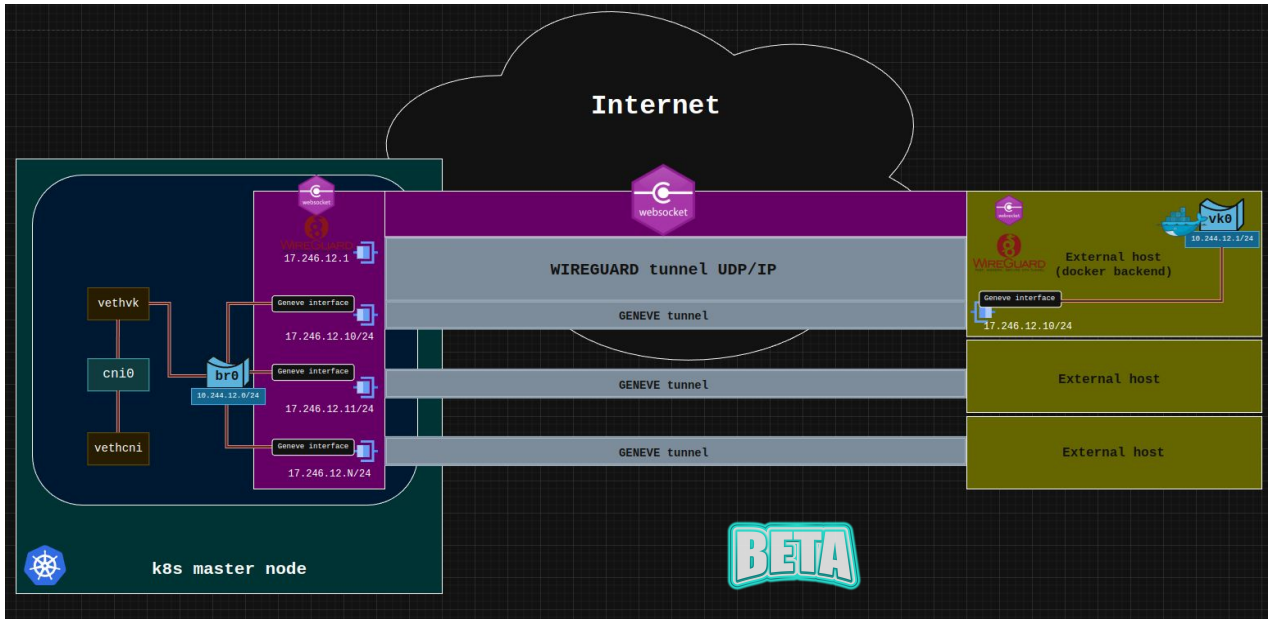
unicore-plugin
https://github.com/interlink-hq/interlink-unicore-plugin

docker-plugin
https://github.com/interlink-hq/interlink-docker-plugin

Sviluppi futuri

Sviluppi futuri - rete

Abilitare InterLink all'uso di una rete pod-to-pod in ambienti eterogenei e distribuiti (HPC, HTC, cloud) tutto a **user level**!



Motivazioni principali

Comunicazione diretta tra pod

Permetterebbe la comunicazione diretta tra pod come in un unico cluster.

Supporto ad applicazioni distribuite

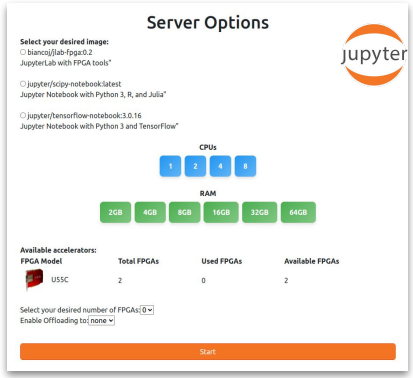
Supporto ad applicazioni con componenti distribuite, come il training distribuito.

Sviluppi futuri - architetture eterogenee (non solo x86)

AI VK è possibile **specificare le risorse disponibili, comprese risorse specializzate, come GPU e FPGA**, permettendo il loro corretto scheduling e gestione tramite plugin remoti, che associano le risorse richieste ai container e ne instradano il carico sull'infrastruttura.

```
virtualNode:
  #image: ghcr.io/dciangot/interlink/vk:v0.17-debug
  image: biancoj/vk:0.39
  resources:
    CPUs: 8
    memGib: 49
    pods: 100
    accelerators:
      #- resource_type: nvidia.com/gpu
      #  model: t4
      #  available: 1
      - resource_type: xilinx.com/fpga
        model: u55c
        available: 2
  HTTPProxies:
    HTTP: null
    HTTPs: null
  HTTP:
    Insecure: true
  kubeletHTTP:
    Insecure: true
```

```
apiVersion: v1
kind: Pod
metadata:
  name: podfpga0
  namespace: interlink
spec:
  restartPolicy: Never
  containers:
  - image: my_fpga_image:latest
    imagePullPolicy: Always
    name: first-fpga-test
    command: ["/bin/bash", "-c"]
    args: ["source /opt/xilinx/xrt/setup.sh; xbutil examine"]
    resources:
        limits:
          xilinx.com/fpga: 1
  dnsPolicy: ClusterFirst
  nodeSelector:
    kubernetes.io/hostname: my-fpga-node
  tolerations:
  - key: virtual-node.interlink/no-schedule
    operator: Exists
    effect: NoSchedule
  - key: accelerator-FPGA
    operator: Equal
    value: U55C
    effect: NoSchedule
```



- ✅ Docker plugin
- Slurm plugin
- HTCondor plugin

- Il plugin Docker supporta già il provisioning delle FPGA.
- È in corso lo sviluppo per abilitare anche gli altri plugin alla gestione di queste risorse.
- **L'obiettivo a lungo termine è estendere il supporto ad altri acceleratori hardware e rendere il sistema compatibile anche con architetture diverse da x86 (RISC-V)**

Summary & conclusions

- Progetto **InterLink** nato circa 2 anni fa nell'ambito dei progetti ICSC e interTwin (EU funded)
- Soluzione **lightweight** per l'integrazione di risorse eterogenee via Kubernetes
- In fase di test con molteplici use case:
  - AI_INFN, analisi ad alto rate, 3D GAN training, …
  - sfruttamento GPU, scale-out su HPC per training/inferenza etc…
- **Uno dei 4 elementi per l'Integrazione tra risorse di** INFN CLOUD CINECA
- Attuali risultati sembrano promettenti → **fase di consolidamento**
- Accettato nella **CNCF Sandbox**: passo chiave per la sostenibilità a lungo termine

CLOUD NATIVE COMPUTING FOUNDATION

Grazie per l'attenzione

Backup

Come:

interLink

HOW?

Estendendo la soluzione **Virtual Kubelet** realizzando un primo draft di un generico API layer per delegare l'esecuzione di POD su **QUALSIASI** backend remoto.

VK core

Un pod che si maschera da nodo e prende in carico le richieste di POD dallo scheduler di K8S

InterLink Server

Si interpone tra il VK e il sidecar. Gestisce le richieste provenienti dal VK e le inoltre al sidecar

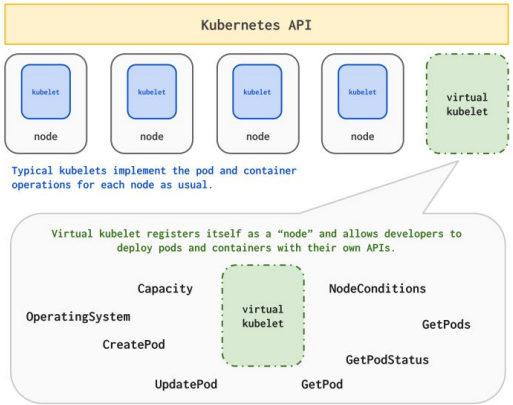
Sidecar

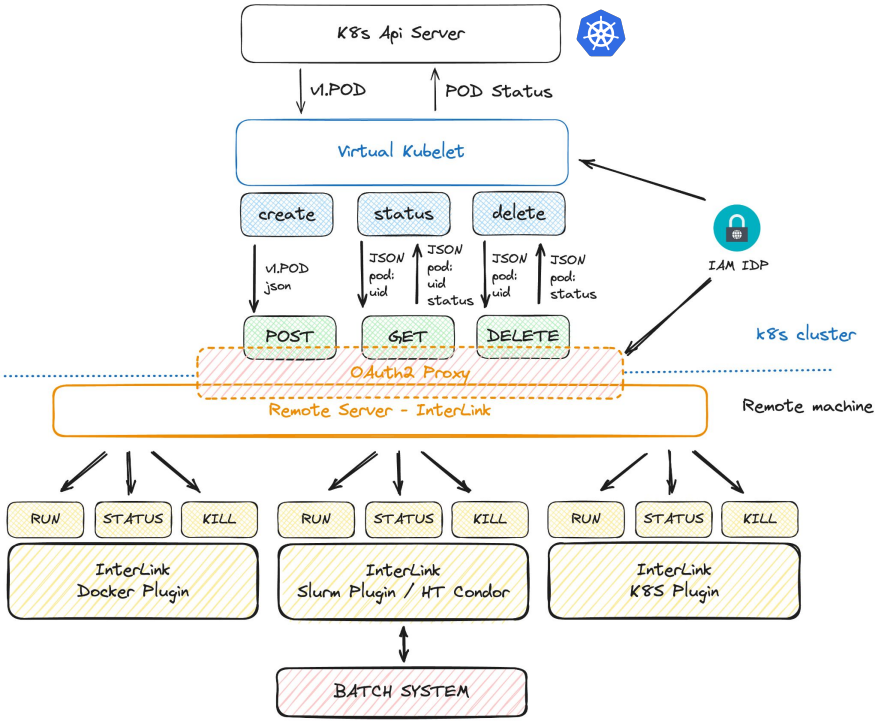
Esegue i container sull'infrastruttura e ritorna il risultato. Comunica con il server InterLink.

WHAT?

Virtual Kubelet

Estendere k8s senza imporre dipendenze specifiche di k8s

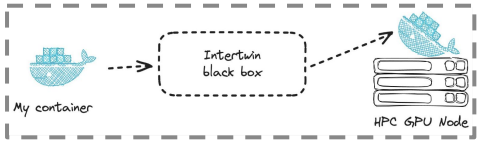




Cosa vogliamo abilitare

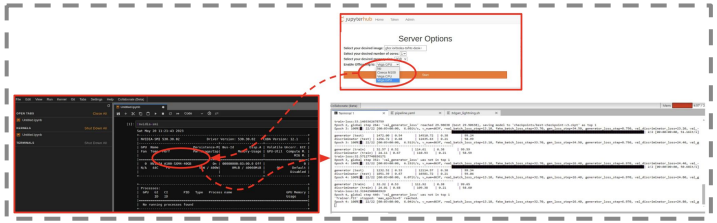
Esecuzione di un POD semplice

Per creare un semplice container e farlo eseguire da un batch slurm remoto in un HPC



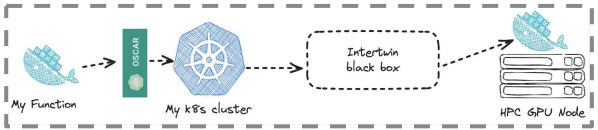
Sessioni interattive

Eseguire istanze JupyterLab su richiesta sia in ambiente HPC sia in ambienti più tipicamente cloud, tramite Kubernetes.



Scale out workload

Per lanciare un payload al verificarsi di un trigger esterno



Gestire backend / modelli di provisioning

- Un insieme unificato di API per integrare risorse eterogenee fornite da provider con tecnologie e architetture differenti

Spostare i payload di un servizio cloud in base alle specifiche esigenze

- payload compute-intensive, memory-intensive, gpu-intensive, …

"nascondere" all'utente l'eterogeneità

- Per l'utente finale l'intero processo è trasparente: il sistema di offloading si occupa di orchestrare l'esecuzione dei workload, decidendo automaticamente su quale backend (Slurm, HTCondor, Kubernetes, ecc.) eseguirli.

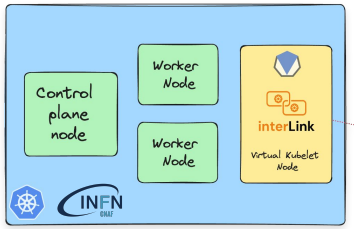
Usare un sistema lightweight e facilmente mantenibile

- Architettura modulare che consente l'integrazione di diversi backend provider attraverso un sistema a plugin

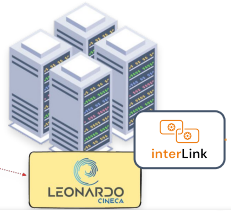
PoC ICSC - Testbed WP6

Ambiente di test WP6 progettato per sfruttare il meccanismo di offloading tramite InterLink, al fine di accedere alle risorse HPC di LEONARDO/Cineca in modo efficiente (**beta users sono BENVENUTI !**).

Confluence *https://confluence.infn.it/x/BgAwK*



Offload

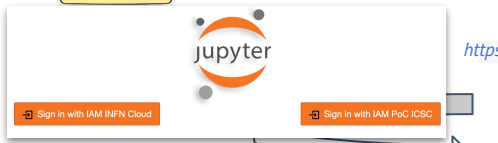


Tutto monitorato

OpenTelemetry

Da semplici POD ...

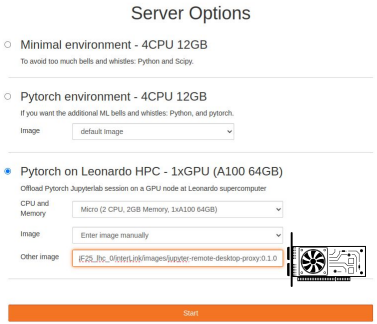
```
apiVersion: v1
kind: Pod
metadata:
  name: helloworld-pod
  annotations:
    slurm-job.vk.io/singularity-options: " --nv --no-mount home,cwd --compat "
    slurm-job.vk.io/flags: " --job-name=helloworld-pod -t 120 --ntasks=1 -A inf25_lhc -p boost_usr_prod --gres gpu:1 "
    slurm-job.vk.io/pre-exec: "export SINGULARITY_USERNS=1; source /etc/profile.d/cvmfs_afuse.sh; "
spec:
  containers:
    - name: helloworld
      image: /leonardo_work/INF25_lhc_0/interLink/images/pytorch-notebook_2025-02-22-build
      command:
        - "bash"
        - "-c"
        - |
          echo 'hello world!'
      resources:
        requests:
          memory: "4Gi"
          cpu: "2"
        limits:
          memory: "8Gi"
          cpu: "4"
  nodeSelector:
    kubernetes.io/hostname: leonardo-virtual-node
  tolerations:
    - key: virtual-node.interlink/no-schedule
      operator: Exists
      effect: NoExecute
    - key: node.kubernetes.io/unreachable
      operator: Exists
```

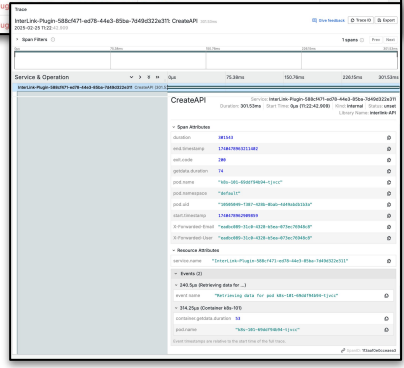


https://jhub.131.154.99.68.myip.cloud.infn.it/hub/home

Fino a JHUB (con autenticazione IAM INFN o PoC-ICSC) per spawn di notebook con **GPU** e **Desktop remoto.**

Qualsiasi altra integrazione (i.e. ray tune, MLFlow etc. supportabile come R/D)





https://monitoring.cloud.infn.it:3000/goto/bKSucbxNR?orgId=1