A scientific perspective on Cloud scalability

Fifth edition of the
Machine Learning @ INFN (ML_INFN)
advanced level hackathon

Diego Ciangottini
Daniele Spiga

Pisa 13-16 Novembre 2023

This Talk

When experiment data grows and process complexity increase laptop or desktop don't fit anymore. At INFN, we recognized the need for a scalable infrastructure-level solution...

One approach can involve the provisioning of on-demand high-level cloud-based services

- Allowing for interactive or batch compute environments
- Exploiting specialized hardware (accelerators, fast disks …) – for AI!

Exploring the feasibility of offloading workloads to "any available resource", i.e. to High-Performance Computing (HPC)

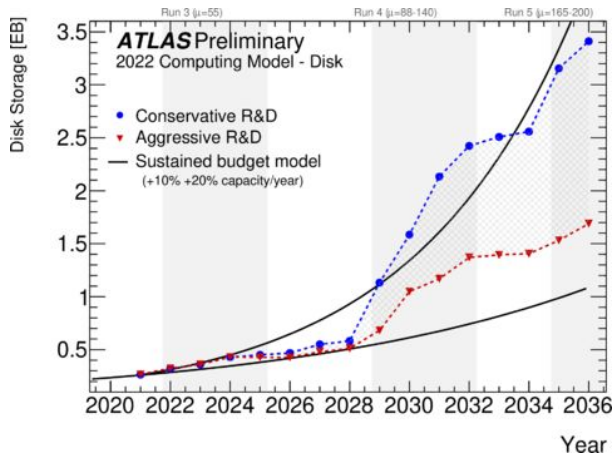
- kubernetes-based technologies

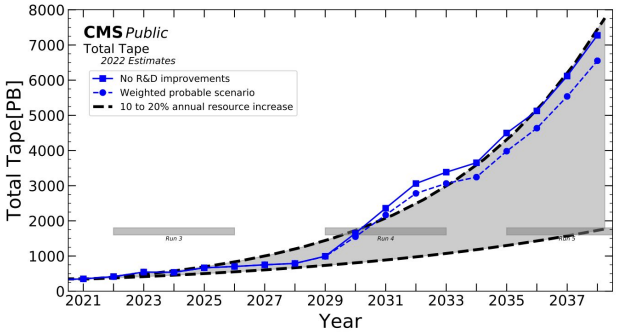
Credits to several people:
- T-Boccali, T. Tedeschi, G.Bianchini, M.Mariotti, M. Sgaravatto…

Just looking at HEP

An example of what we expect *circa 2030*

- This is the “optimistic” (in the IT sense) view:
  - Complete CMS (one of the CERN LHC experiments) collecting 50B events per year, plus 100B Simulated events; at the smallest format (10 kB/event), 1500 TB
  - One physics analysis ~ 20% of the full dataset → 300 TB
  - This needs to be fed to ML-DL analysis-specific systems for training
- This is the “pessimistic” (in the IT sense) view:
  - We have a full end-to-end GNN based reconstruction algorithm, fed with raw data (10 MB/event)
  - We need to feed it with 1.5 EB/y while taking data and performing simulations





Our strategy at INFN

Allow researchers to exploit "free" and open services to manage workflows, build pipelines, data processing and analysis and, of course, **to share/to reuse technical solutions**

- Allow researchers to focus on science

Technical drivers:

- to enable users **to create and provision infrastructure deployments, automatically and repeatedly**, **with almost zero effort**.
- To Implement the ***Infrastructure as Code*** **paradigm** based on declarative approach: **allows to describe "What" instead of "How"**
    - Let the underlying system to deal with technicalities
- To promote (and support) **container-based solutions**
- To grant data sharing among users/infrastructures

spiga@pg.infn.it

…and from user perspective: few pillars

end users should handle just few pillars

- What the user should/might see out of all of the underlying system?

Software management: a central role is played by container. A standard unit of software suitable to create **user tailored environment**, (share and port everywhere).

- Users create containers, the system distribute them via global file systems…

Infrastructure management: in principle user might chose to know “nothing” about infrastructure (SaaS model and above).

- If a researcher need/swants to customize its infrastructure, the system (the Cloud) should offer handles… **through templates** [see later]

spiga@pg.infn.it

Don’t forget about Data

(hence the DataCloud name)

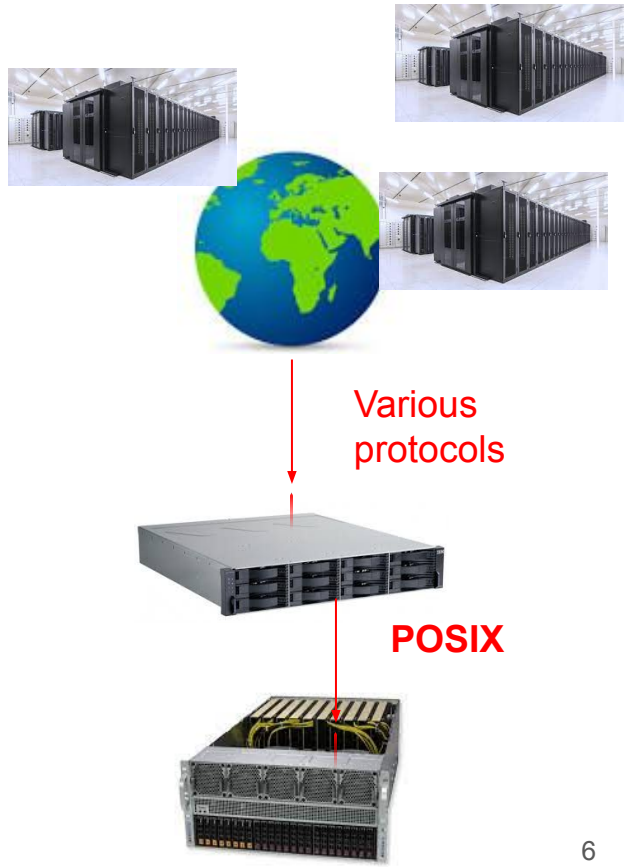
Building on top of WLCG Data Lake model (via FTS, Rucio etc)

We foresee an infrastructure system that offers handles to manage potentially large volumes of data

- Seamlessly **transfers data** into or out of storage services, **including those provided by HPC**

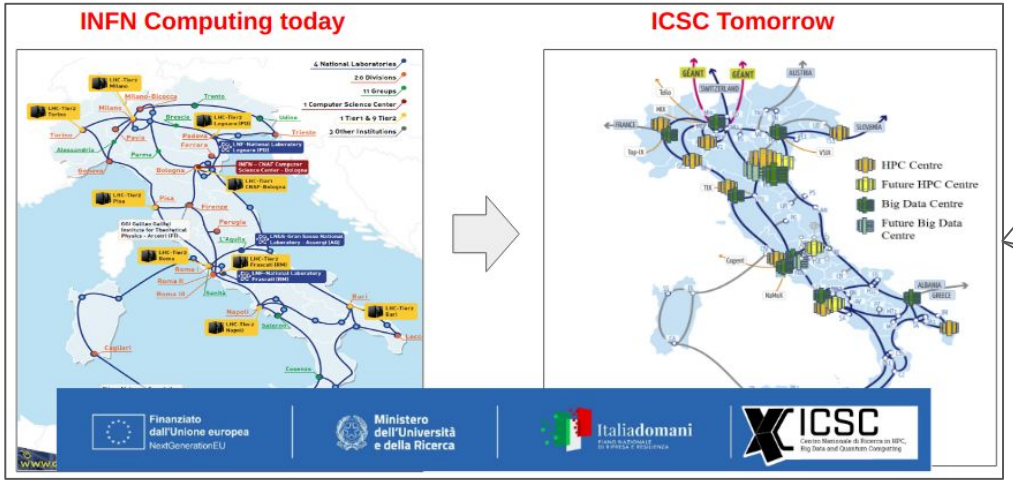
Data Access to possibly remotely located data at high speed is essential for AI and Big Data

- Exploiting fast disk (NVMe) to implement **cache systems on computing node**
- Providing POSIX-like API for accessing storage (local/remote)

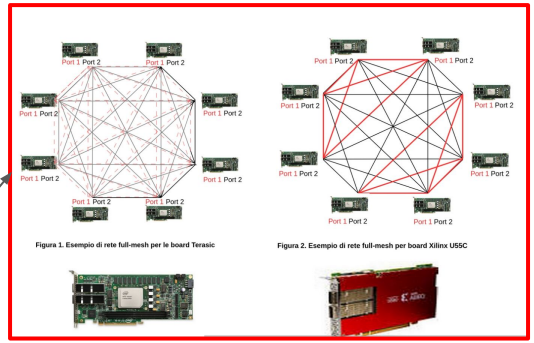


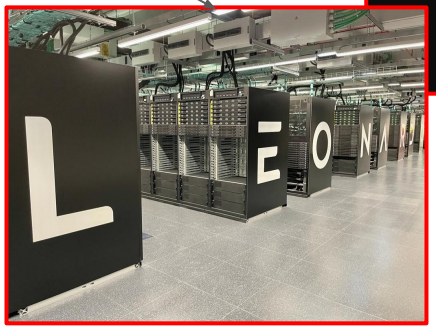
Do we really need “scalability”?

A first look from the infrastructural perspectives



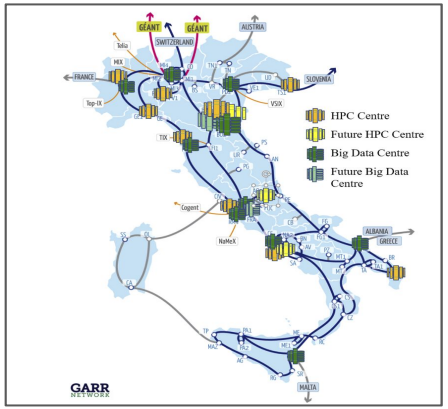
FPGA Clusters





The vision

We envision a model where **high level services deployed on a cloud provider can be enabled to transparently execute containerized payload everywhere,** ie on a remote batch system such as a SLURM on a HPC system or over “fat nodes”



We want to implement the "continuum" in a heterogeneous context

Where do we start? Enabling the **payload offloading**

- A service instantiated and running at provider ***P1*** extends transparently to the user, on provider ***P2*** for processing a given Workload

There are few assumptions behind such a model :

- **Edge services, runtime environment, networking solutions…**

The offloading in a nutshell

Transparently extend “*any* **application** *anywhere*”

- **To federate (highly) heterogeneous and disparate providers**
  - **enabling a "transparent payload offloading"**

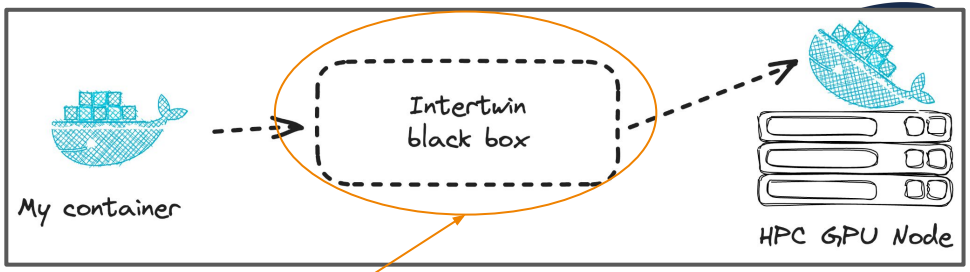
Any containerized application

To delegate the containers execution

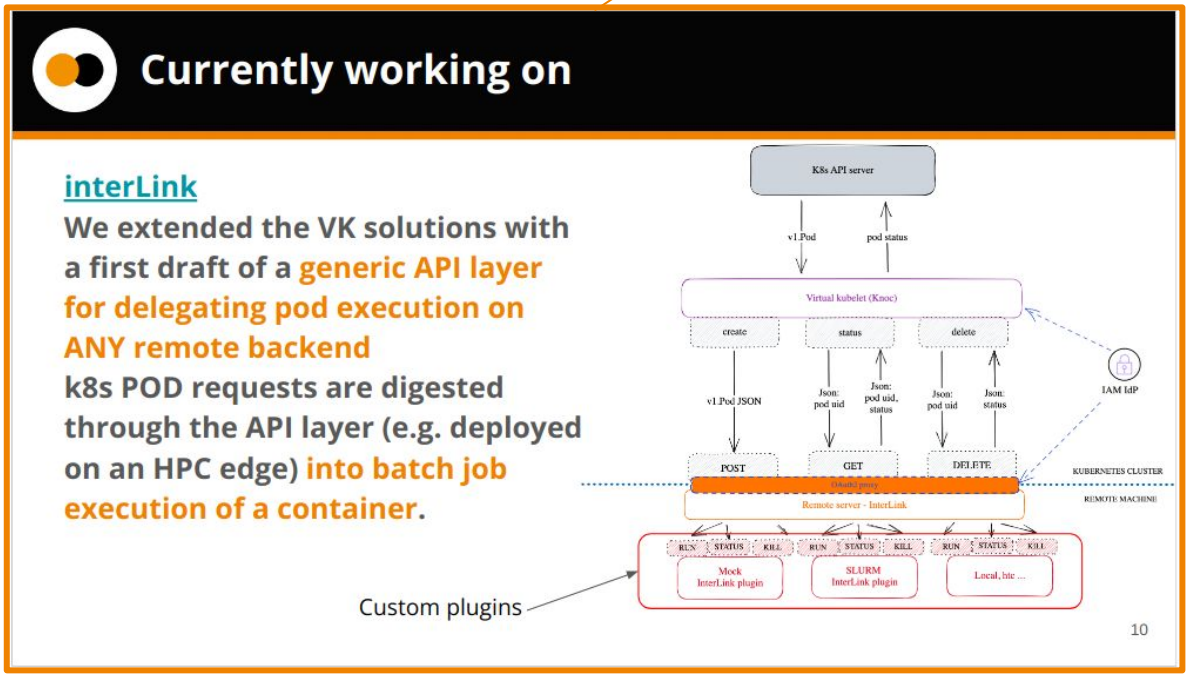
The user experience

The interLink project

(aka who does the magic)



Extends the container orchestration de-facto standard (K8s) to support offloading under the hood



interTwin

interTwin EU collaboration

Scalability: scientific perspectives

(Not only infrastructure)

Easily process data access exploiting huge amount of (possibly remote) computing capacity (in a small amount of time)

- Datasets to analyze is (much) larger that your memory (RAM)
- We would like to use all available computing power on local machine or on many (possibly remote) different machine
  - : time to insight; high rate; interactivity

Specialized HW GPUs/FPGA and fast disk

- Development of ML pipelines, Jupyter notebooks

Opportunistic usage of computing Capacity

- A process can easily run in background over temporary unused resources

spiga@pg.infn.it

Interactive processing in a distributed system

(R&D on analysis at CMS) [see here]

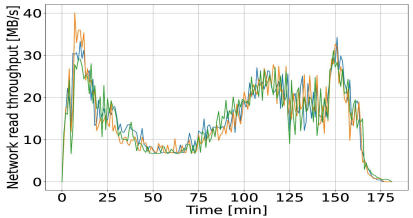
	Legacy	RDF
Overall time [min]	181 ± 1	23.8 ± 0.6
Overall rate [events/s]	60.5k ± 0.3k	465k ± 11k
Job rate [events/s]	786 ± 12	6915 ± 35
Job event-loop rate [events/s]	858 ± 14	7632 ± 34

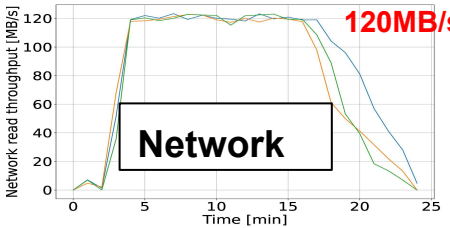
R&D on analysis at High Luminosity LHC

- optimizing the computing and storage resource utilization

Testing software featuring a declarative programming model and interactive workflows

- Increasing data processing **throughput** is crucial
- Fast Turnaround Reducing analysis "**time to insight**"





All this impact on infrastructure: need prototype resources integration models to efficiently leverage computing capacity

- Integrate already deployed (grid) infrastructure
- Transparently access specialized HW
- Scale toward opportunistic (cloud/HPC)

Prototyping integration models

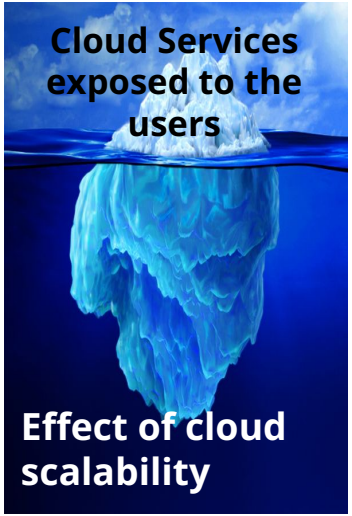
(any resource anytime)

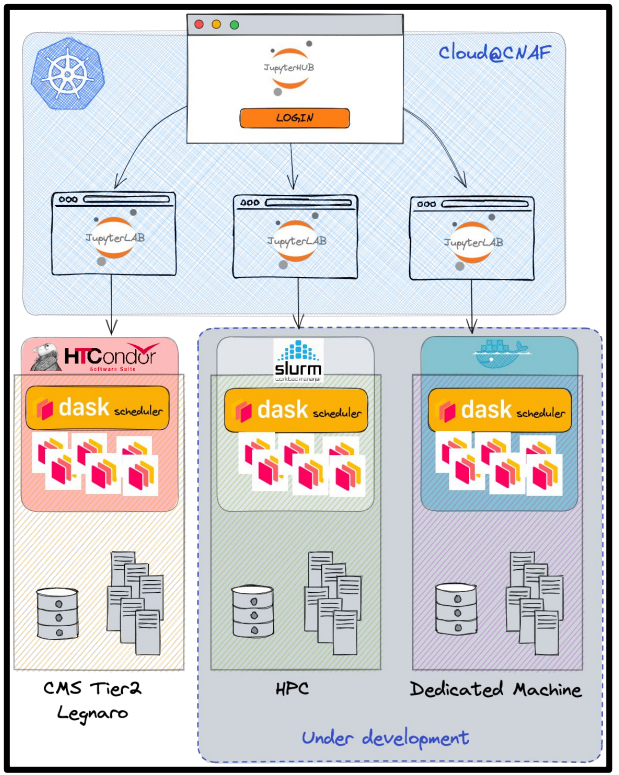
A single entrypoint (HUB) for the data analysis deployed on Kubernetes

- **Containerization to allow user to customize their runtime environment**
- Jupyterlab environment:
  - Both “a-la-batch” and interactive processing allowed

Integration of heterogeneous resources under the same pool:

- Existing WLCG infrastructure and batch-systems for interactive use used for both legacy and interactive processing:
  - **Using an HTCondor overlay and Dask in HTCondor mode**

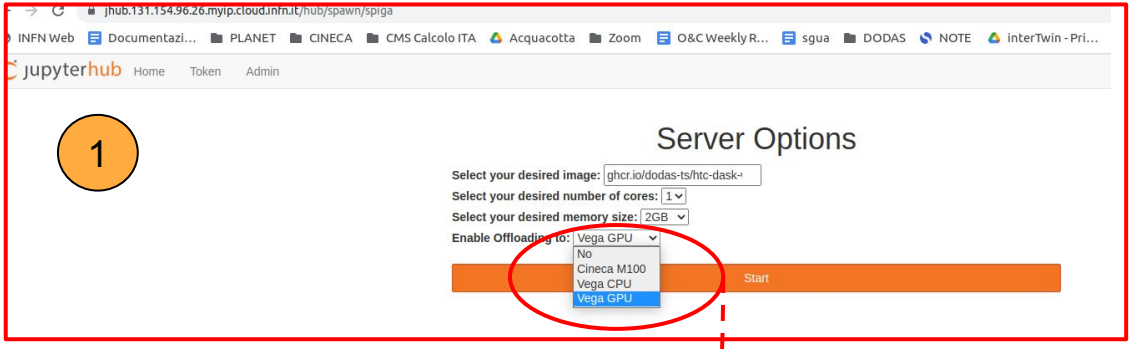


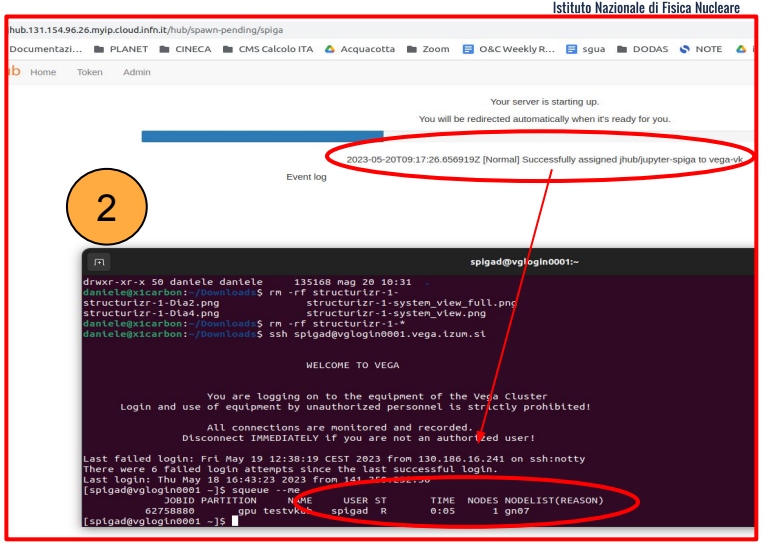


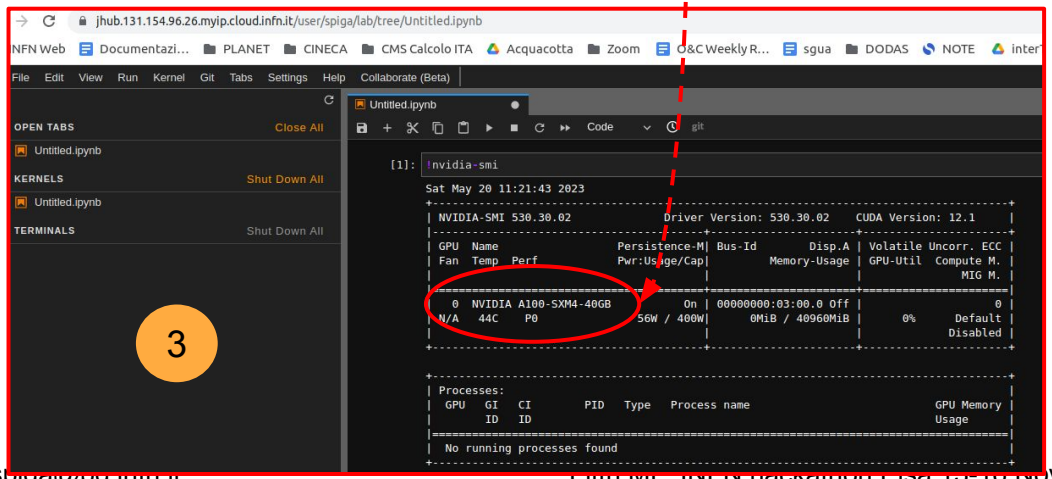
Specialized HW

(aka a EuroHPC under my notebook)

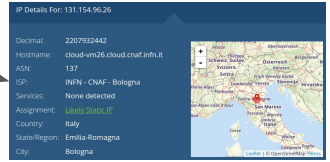
Istituto Nazionale di Fisica Nucleare





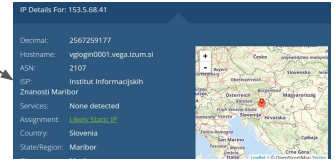


1. **JupyterHub @ CNAF (su k8s)**

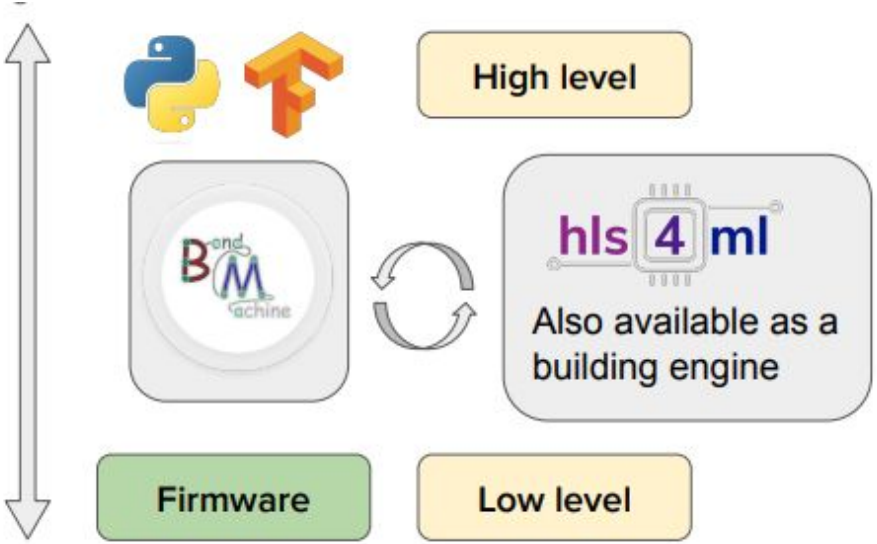


2. **Jupyter Lab @ Vega (via slurm)**

3. **Accesso user-level a GPU (A100) available on Vega WN**



FPGA and Cloud: enabling a declarative access



- Starting from high-level code and standard ML framework,
- with HLS tools like BondMachine and hls4ml, get the firmware
- implementations of machine learning algorithms

Training courses like this one

Are there concrete opportunities?

FPGA Clusters expected by 2024

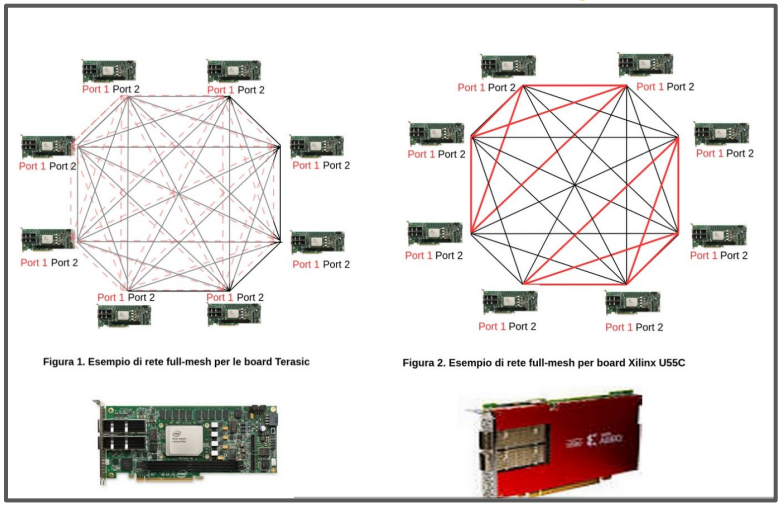


Figura 1. Esempio di rete full-mesh per le board Terasic

Figura 2. Esempio di rete full-mesh per board Xilinx U55C

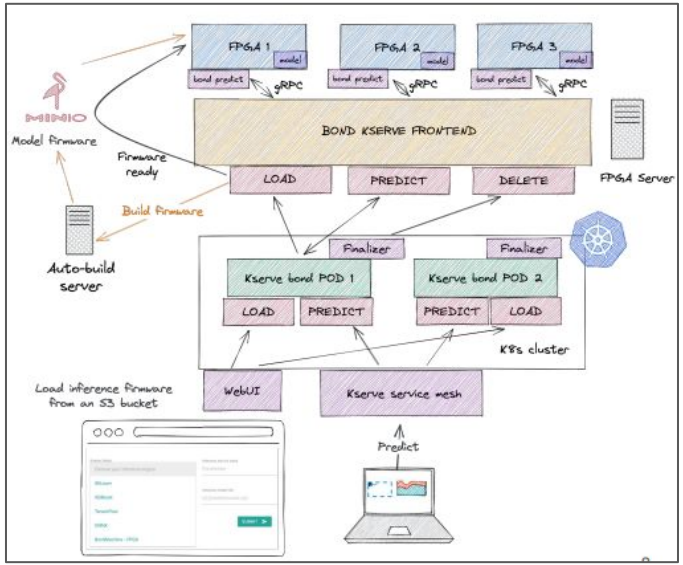
4 server 4U (2xEpyc 7313, 4 slot pci4, Ram 1024GB , 2xEthernet 10Gbps),

- 2 server con 4 board Xilinx U55C
- 2 server con 4 board Terasic
  DE10-Agilex Development Board (Part Number P0701)

per un totale di 8 board Xilinx U55C e 8 board Terasic.

2 **Storage:**
- 10 TB per i server U55C in Raid5 + (SSD NVMe)
- 10 TB per i server Agilex in Raid5 + (SSD NVMe)

Automatically synthesize FPGA firmware from a generic ML model **and serve it through a cloud native system**

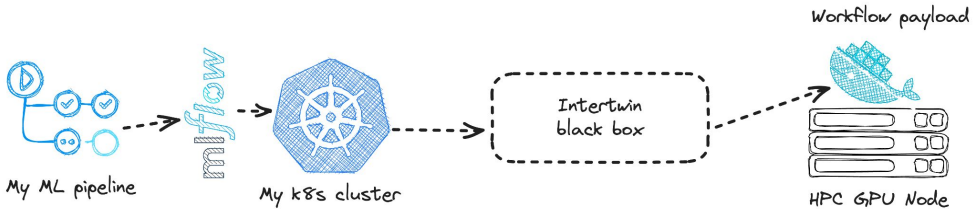


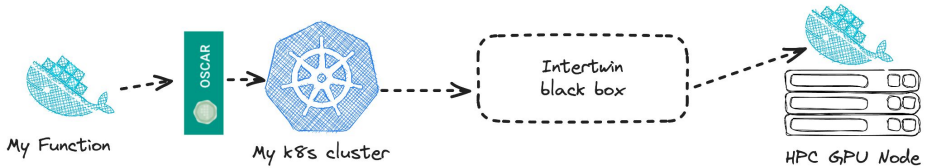
Other advanced R&D

(collaboration with CERN and UPV)

Frameworks for **DAG/workflow managements are usually well integrated with K8s APIs**

- Airflow/Kubeflow pipelines/Argo wokflows/MLFlow are no exceptions





Serverless: executing payload in response of an external trigger

- Being it a storage event or a web server call

Summary

Cloud native technologies can be a key to a successful computing model implementation

- As you saw today the DataCloud project is evolving “toward a cloud oriented system”
- INFN is very active in developing a wide ecosystem where researchers can exploit solutions, do practice and share/reuse solutions

The current one it’s a very hectic period with a lot of forces and inputs

- Few initiatives have been sketched as examples
- You might have project that could benefit from what we discussed today
  - Feel free to contact us

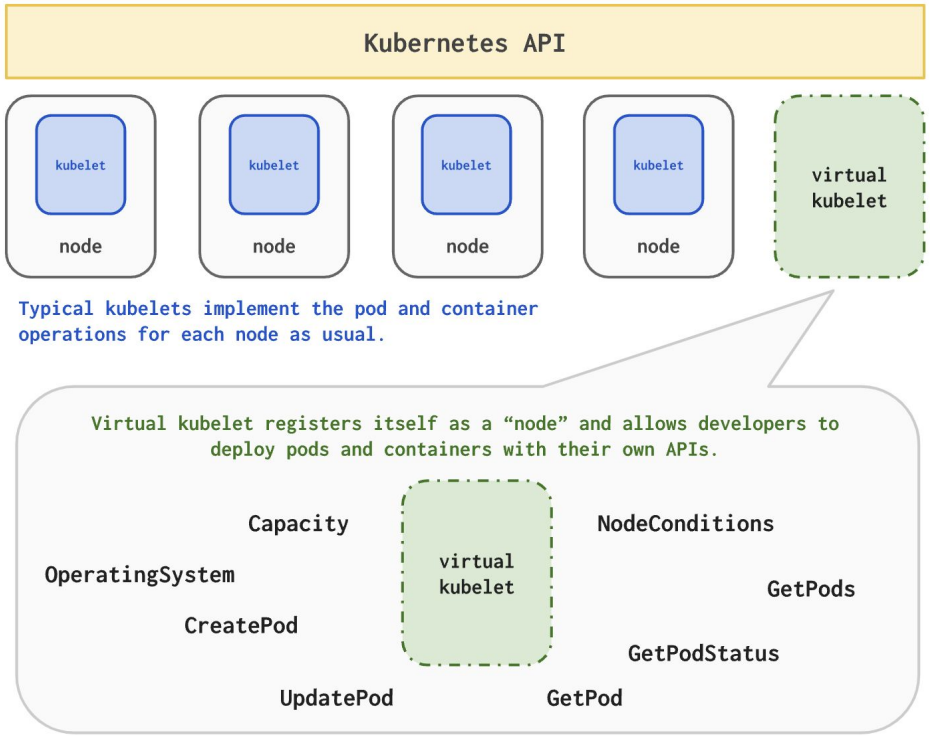
spiga@pg.infn.it

BACKUP

The enabling technology

Virtual kubelet (VK):

“Open-source Kubernetes kubelet implementation that masquerades as a kubelet. This allows Kubernetes nodes to be backed by Virtual Kubelet providers”



All in all the flow we want to achieve