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

How to support advanced use cases: Technological highlights

Diego Ciangottini - INFN

Workshop on "Quasi-Interactive Analysis of Big Data with High Throughput"

Where did we start?

In the past days you saw examples of the power of the new and distributed-friendly frameworks.

Especially for interactive analysis

What about cases that goes beyond interactivity?

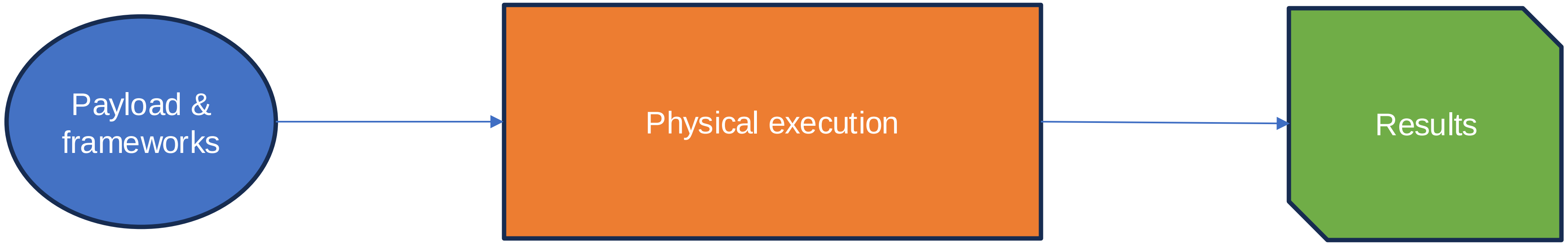
Let's do a step back now.. Trying to generalize

Untangling the common requirements

If you step back enough you should start seeing some common needs for "any" kind of analysis: a payload to be executed on the optimal set of resource

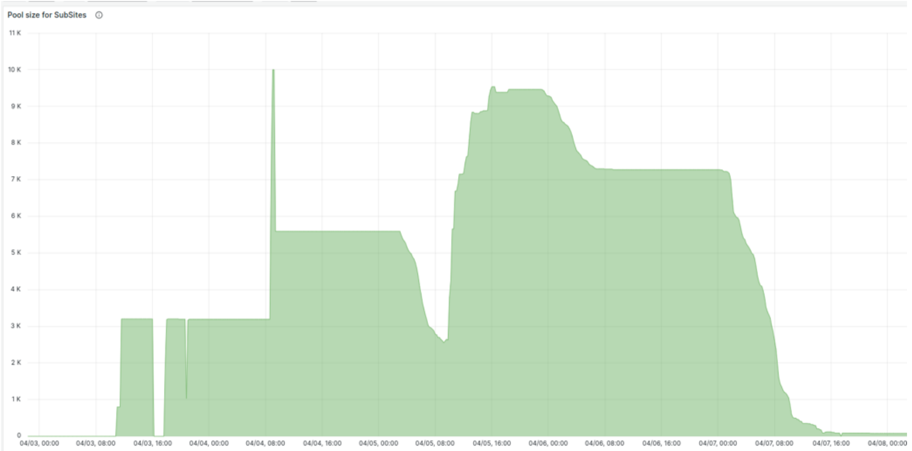
Wearing the right glasses, we can also spot that all of them can be "easily containerizable"

Good, now we have a set of containers to be executed for different use cases…

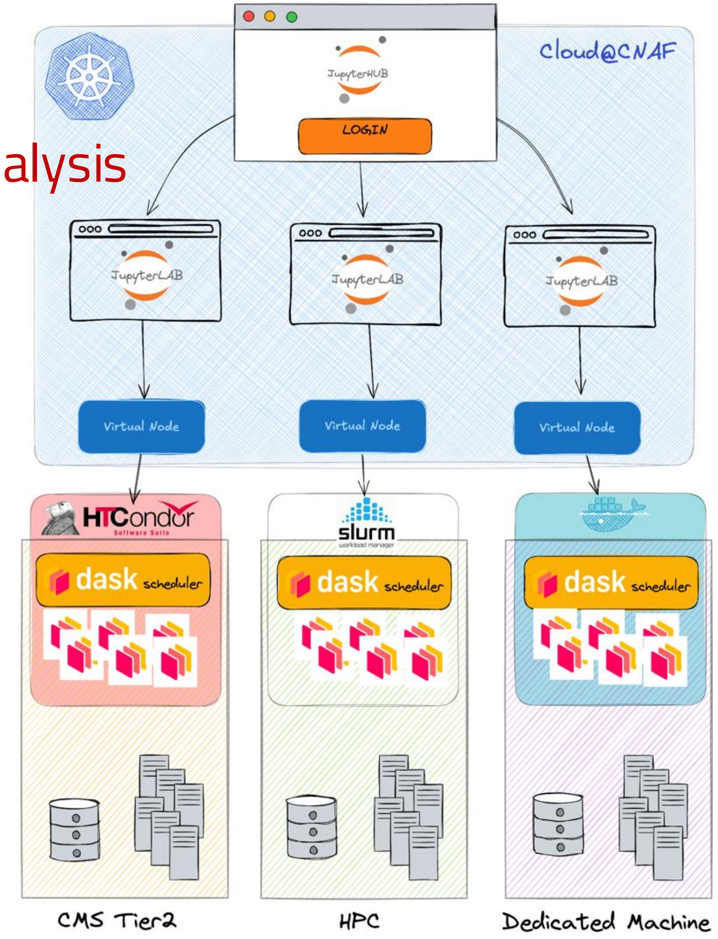


We do have a certain experience behind us on frameworks

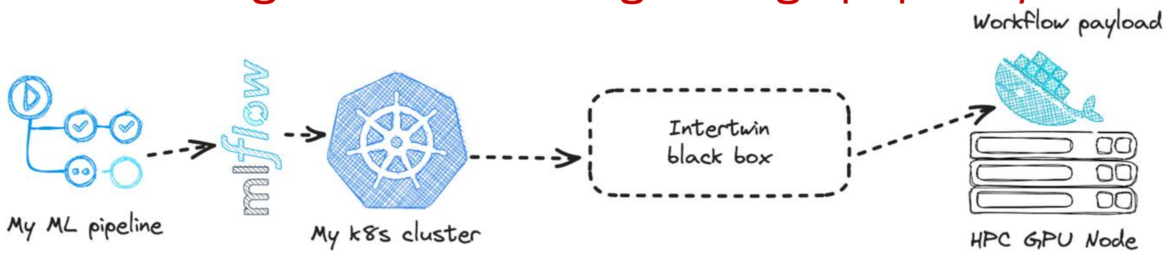
Grid/Batch processing



Interactive analysis



ML/AI training and inference (growing up quickly)



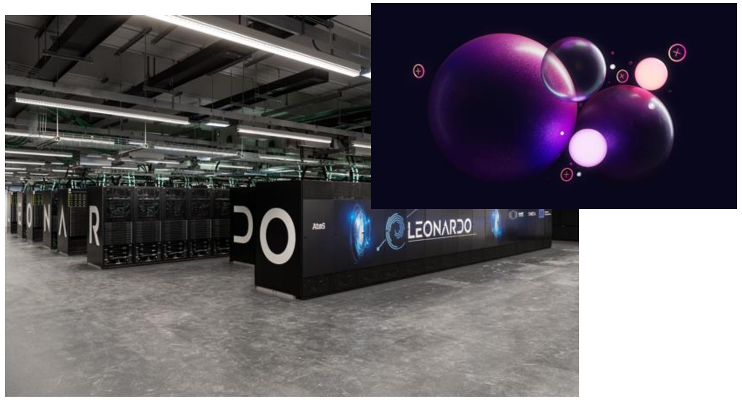
Physical execution

Same for resource provisioning

We are used to not so unusual problem of making the matching with the payloads.

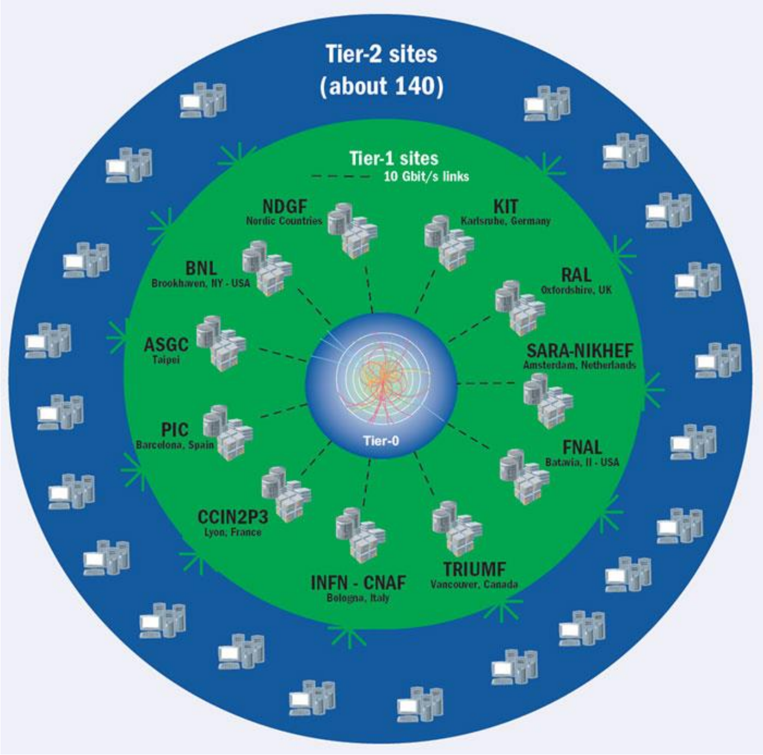
Can we possibly abstract this with a single interface?

Supercomputing at HPC
+
HPC bubbles



Cloud-like resources:
on-demand GPUs or beefy VMs

Grid resources



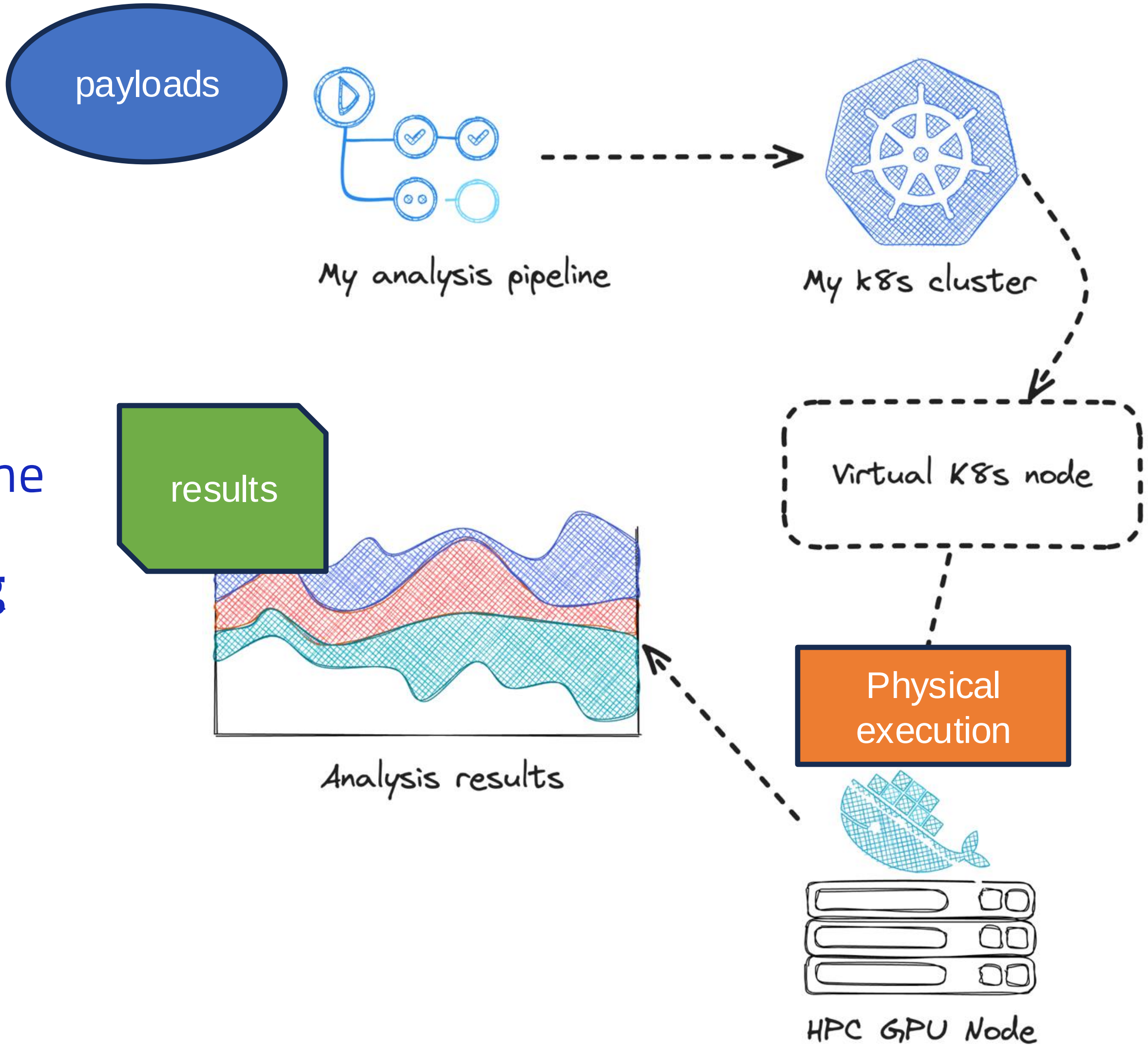
In a nutshell

Now that there is a de-facto standard for orchestrating containers (and not only) → Kubernetes

Can we think of exploiting advanced use cases and framework to test a "pure" Kubernetes approach for some CMS analysis needs?
Other then "just" the interactive lab? **Like for everything**

Why?

1. Cloud-native community ecosystem compatibility
2. Interoperability (buzzword!)
3. Declarative approach extended to the resource request!
4. Git-Ops native analysis?



Road to a cloud native analysis (a possible evolution path)

Let’s try if we can use a container orchestrator to orchestrate… containers

Of course, in a special way, ”offloading”* the execution to external endpoints

A user should not ever notice it! Just assigning their pods to a special cluster node where the magic happens!

Take home message: “We are kind of convinced to be ready for giving it a try”

"Offloading" refers to the process of delegating the execution of a container to a remote resource instead of a physical node in your cluster.

Analysis/Training/Whatever in a Pod

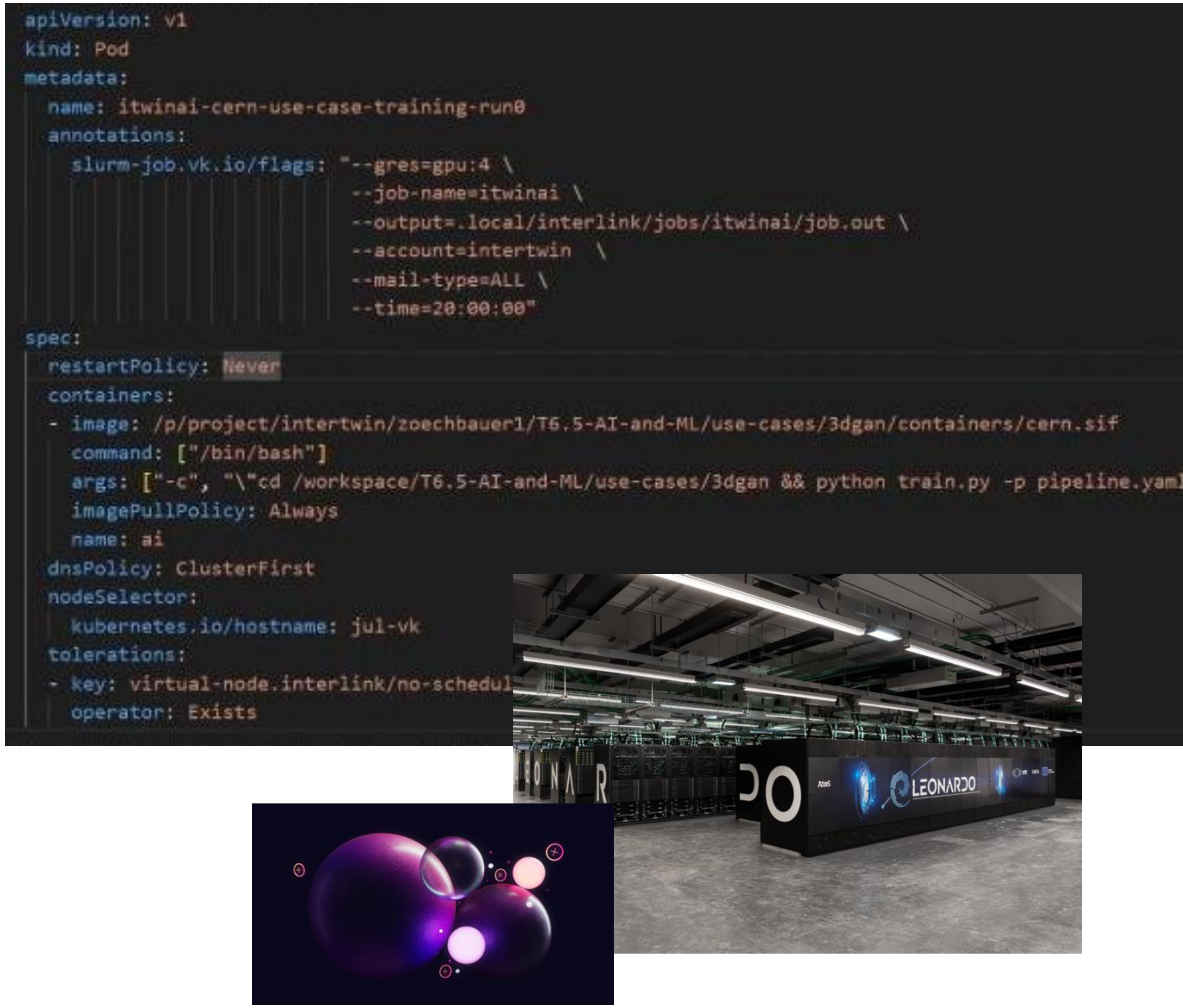
When it makes sense?
You have tasks that can run in a “standalone” container.

Why?
You can reproduce it everywhere on any cloud Maybe accessing GPUs/resources with the least effort

How?
You need to practice Kubernetes basics.
Pods spec is just a way to configure container execution

Is it enough?
ML/AI training and inference are an obvious use case, but also there, ancillary services might need some setup

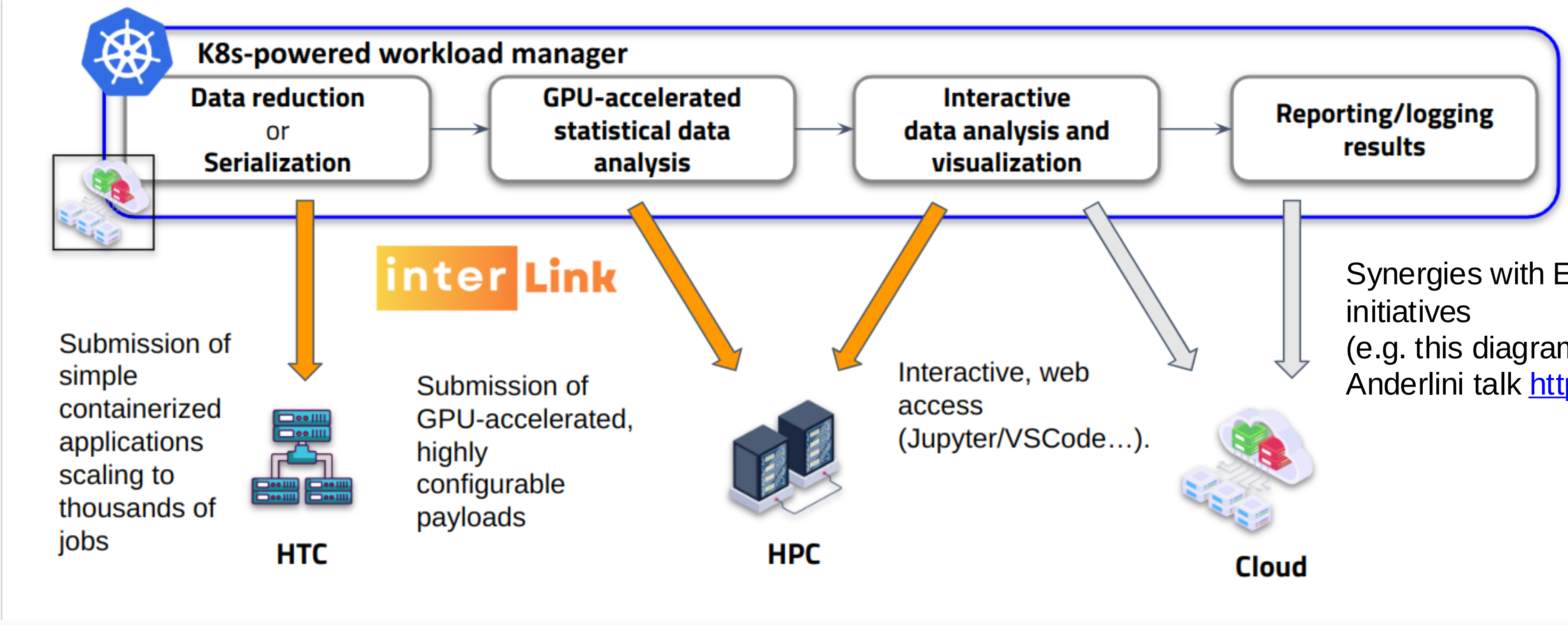
See Raffaele’s next presentation



```
apiVersion: v1
kind: Pod
metadata:
  name: itwinai-cern-use-case-training-run0
  annotations:
    slurm-job.vk.io/flags: "--gres=gpu:4 \
                            --job-name=itwinai \
                            --output=.local/interlink/jobs/itwinai/job.out \
                            --account=intertwin  \
                            --mail-type=ALL \
                            --time=20:00:00"
spec:
  restartPolicy: Never
  containers:
  - image: /p/project/intertwin/zoechbauer1/T6.5-AI-and-ML/use-cases/3dgan/containers/cern.sif
    command: ["/bin/bash"]
    args: ["-c", "\"cd /workspace/T6.5-AI-and-ML/use-cases/3dgan && python train.py -p pipeline.yaml
    imagePullPolicy: Always
    name: ai
  dnsPolicy: ClusterFirst
  nodeSelector:
    kubernetes.io/hostname: jul-vk
  tolerations:
  - key: virtual-node.interlink/no-schedul
    operator: Exists
```

Matching different resources to different payloads… with one API

Evaluate the whole chain with CMS requires validation of each one of these steps with real use cases



K8s-powered workload manager

Data reduction or Serialization

GPU-accelerated statistical data analysis

Interactive data analysis and visualization

Reporting/logging results

interLink

Submission of simple containerized applications scaling to thousands of jobs

HTC

Submission of GPU-accelerated, highly configurable payloads

HPC

Interactive, web access (Jupyter/VSCode…).

Cloud

8

Synergies with EU projects (interTwin) and INFN initiatives
(e.g. this diagram is courtesy of M.Barbetti and L. Anderlini talk https://indi.to/gqWcc)

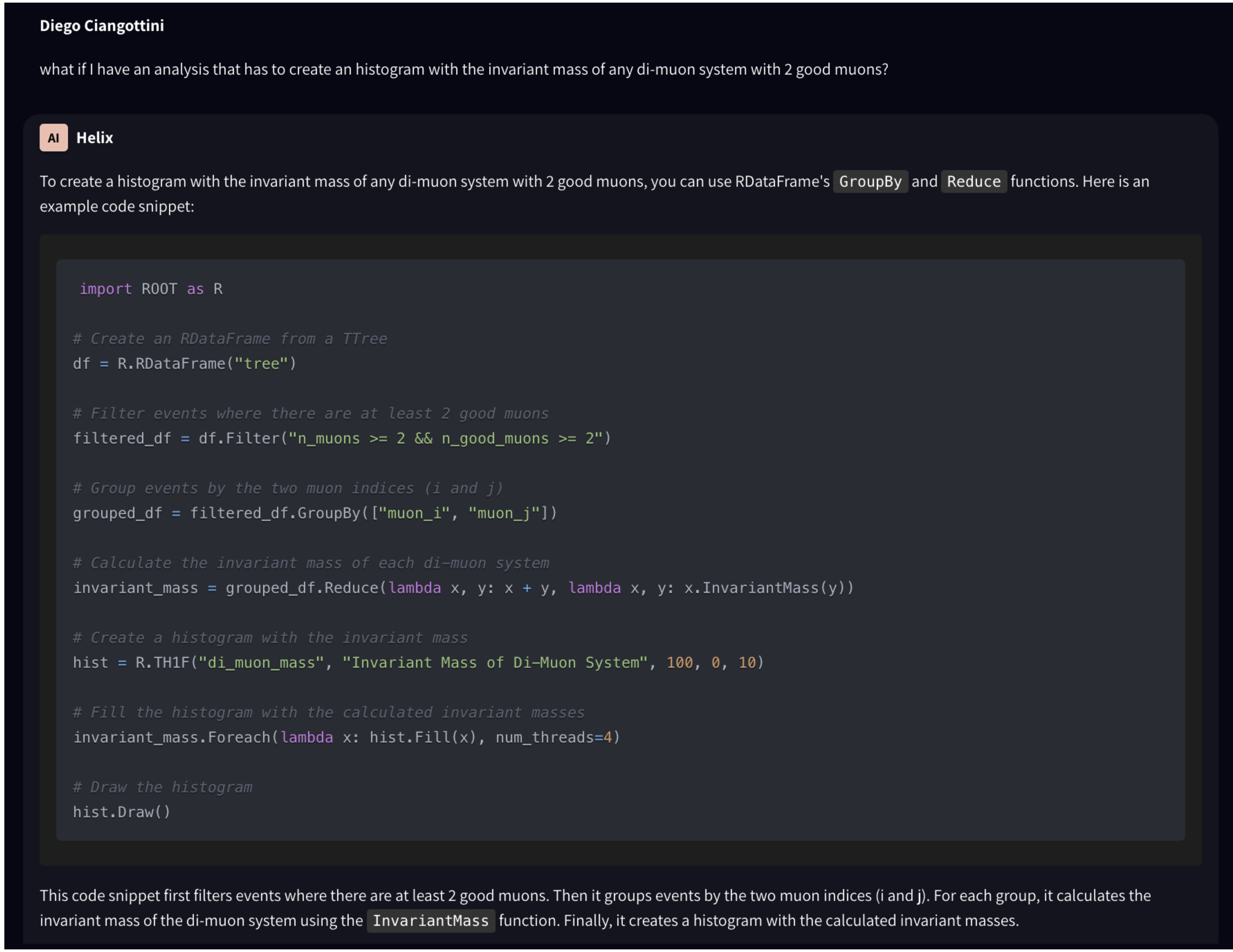
Much more has yet to come…

Other use cases might be just around the corner.

Did you say inference?

Oh no, you said LLM/transformer…

Yes, you can develop an AI app seamless on supercomputers too! Thanks to the very same enabling technology



Diego Ciangottini

what if I have an analysis that has to create an histogram with the invariant mass of any di-muon system with 2 good muons?

Helix

To create a histogram with the invariant mass of any di-muon system with 2 good muons, you can use RDataFrame's `GroupBy` and `Reduce` functions. Here is an example code snippet:

```
import ROOT as R

# Create an RDataFrame from a TTree
df = R.RDataFrame("tree")

# Filter events where there are at least 2 good muons
filtered_df = df.Filter("n_muons >= 2 && n_good_muons >= 2")

# Group events by the two muon indices (i and j)
grouped_df = filtered_df.GroupBy(["muon_i", "muon_j"])

# Calculate the invariant mass of each di-muon system
invariant_mass = grouped_df.Reduce(lambda x, y: x + y, lambda x, y: x.InvariantMass(y))

# Create a histogram with the invariant mass
hist = R.TH1F("di_muon_mass", "Invariant Mass of Di-Muon System", 100, 0, 10)

# Fill the histogram with the calculated invariant masses
invariant_mass.Foreach(lambda x: hist.Fill(x), num_threads=4)

# Draw the histogram
hist.Draw()
```

This code snippet first filters events where there are at least 2 good muons. Then it groups events by the two muon indices (i and j). For each group, it calculates the invariant mass of the di-muon system using the `InvariantMass` function. Finally, it creates a histogram with the calculated invariant masses.

Do we have the full set?

We do want to understand/demonstrate if we can deliver what I just presented for CMS analysis

Where are the bottlenecks, maintenance costs, limitations, etc..

This are the kind of things for which experience with real-world use cases is key at the moment!

In other words..

Call to action

We know that we can validate/refine/retune our hypotesis with CMS analysis users
see Raffaele example in a few mins

Things are ready to support new/advanced use cases:
1. Reach out for accessing resources
2. Provide feedback
3. Try again and repeat

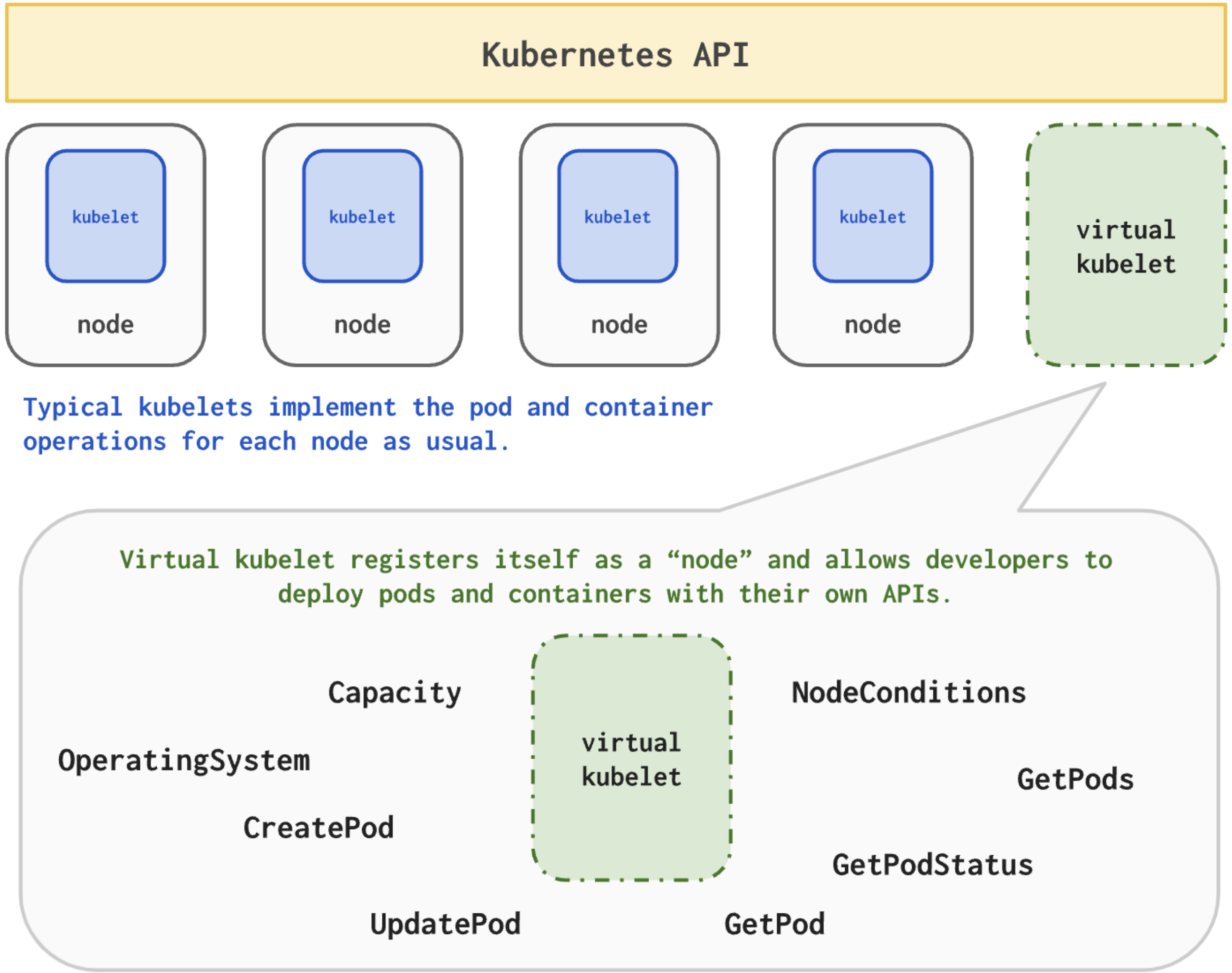
Of course, open to ANY more detailed feedback from Physics goal and requirements

If I had to sell it, I'd tell you that in this way you can get extra resources for free… plus learning tools and techniques that are cutting edge.

Just some final references for geeks

https://intertwin-eu.github.io/interLink/

interLink


Kubernetes API
kubelet
node
kubelet
node
kubelet
node
kubelet
node
virtual kubelet
Typical kubelets implement the pod and container operations for each node as usual.
Virtual kubelet registers itself as a "node" and allows developers to deploy pods and containers with their own APIs.
Capacity
OperatingSystem
CreatePod
UpdatePod
virtual kubelet
NodeConditions
GetPods
GetPodStatus
GetPod