

Scalable Machine Learning with Kubeflow at CERN

Preparation, Training and Model Serving

Dejan Golubovic, Ricardo Rocha
CERN IT-PW-PI

Contents

Motivation

Infrastructure

Kubeflow Features

Demo

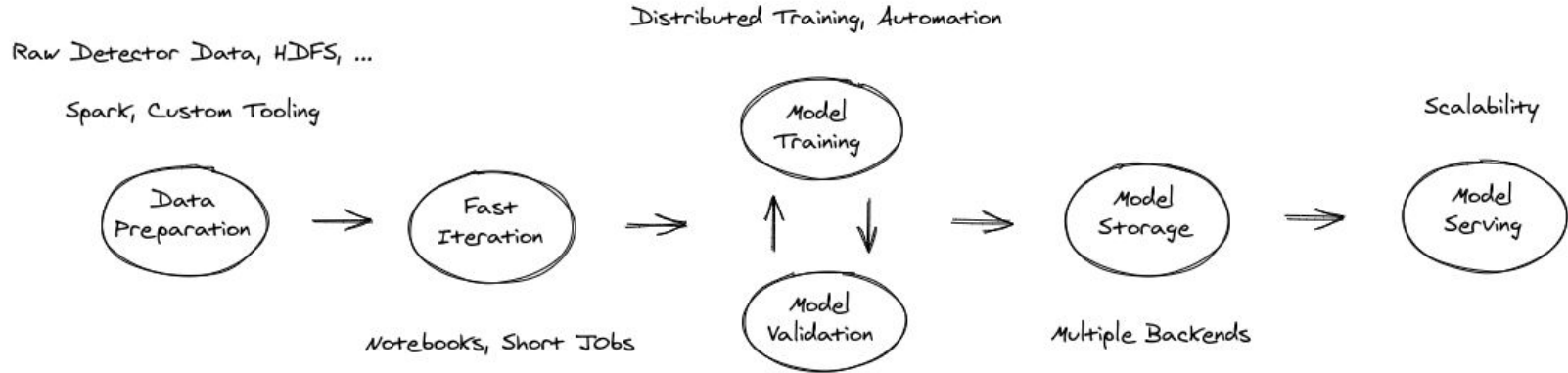
Resource Management

Use Cases

Conclusions

Motivation

Offer a platform to manage the full machine learning lifecycle

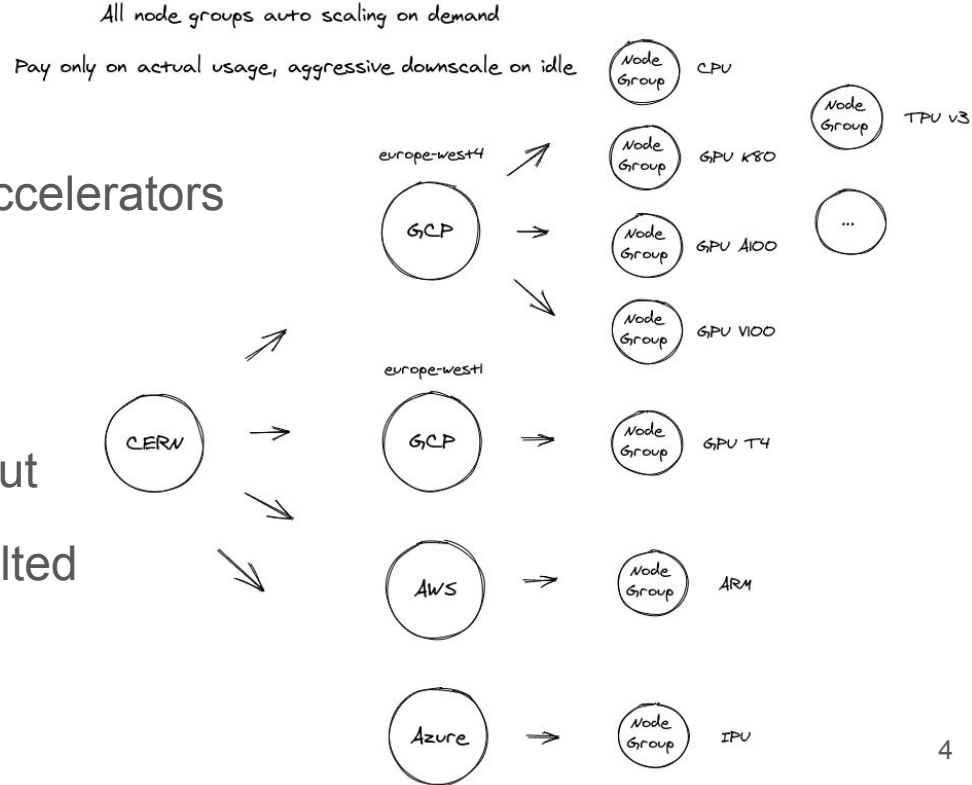


Motivation

Ensure **efficient usage** of our
on premises GPU resources

Provide easy access to **public cloud** accelerators
(**GPUs, TPUs, IPU**s, FPGAs, ...)

Bursting setup already demonstrated, but
access to resources temporarily halted



Infrastructure



Based on [Kubeflow](#), the machine learning toolkit for Kubernetes

Open source project started by Google in 2017

Applied to become a [CNCF incubating](#) project in October 2022

Declarative API, operators, auto healing, application and cluster auto scaling

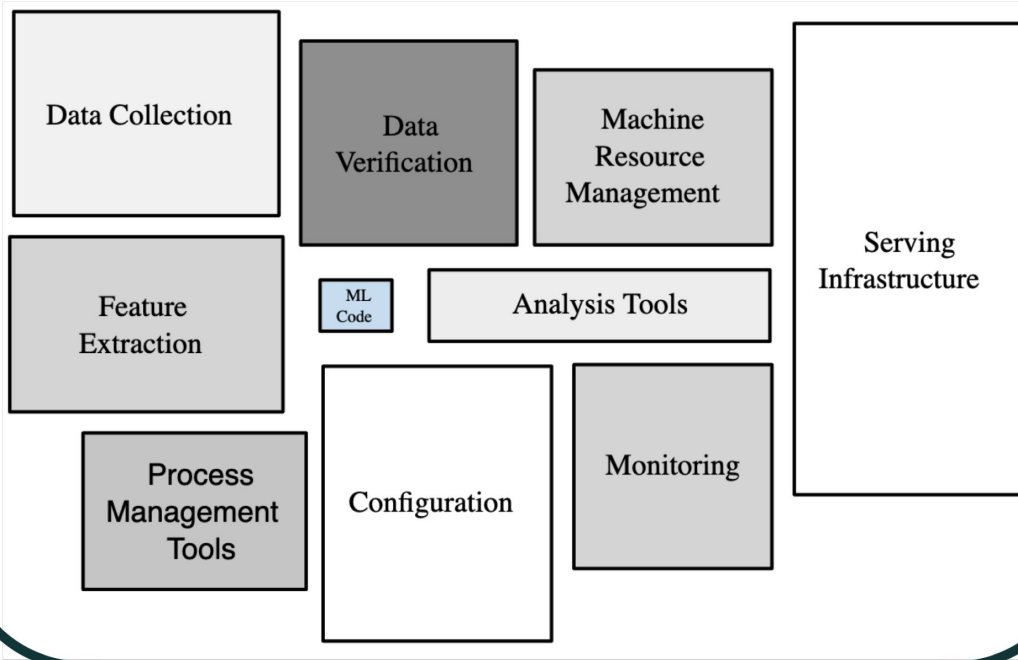
Support for most common frameworks (**TensorFlow**, **PyTorch**, MXNET, ...)

In production at multiple companies

Google, Spotify, Bloomberg, Zillow, Arrikto, Aurora...



Kubeflow



Kubeflow Components and Features

Notebooks



Machine Learning Pipelines

AutoML - Hyperparameter Optimization

Distributed Training

Tensorboards

Model Serving



Notebooks

Easiest way to **start experimenting** with Kubeflow

Integration with other Kubeflow components

Pipelines, distributed training, inference, AutoML

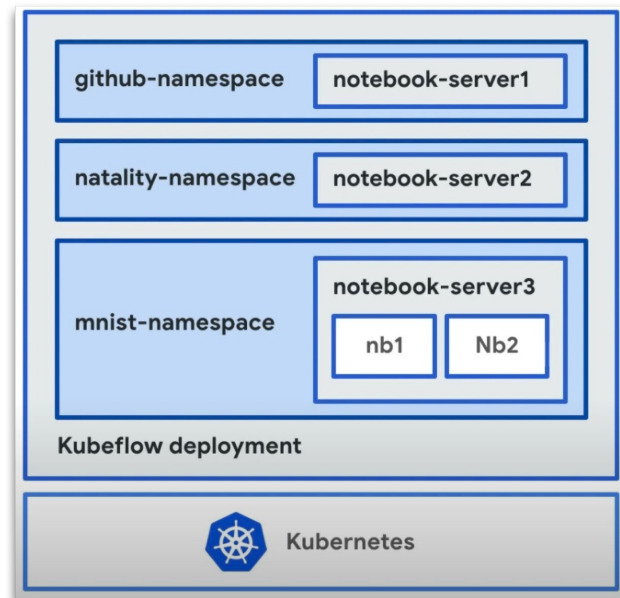
Ability to **customize Python environment**

Or use prebuilt images (Tensorflow, Pytorch)

Select resources (CPU, MEM, GPU)

Pause notebook servers if idle for too long

Good for **experimentation and prototyping phase**



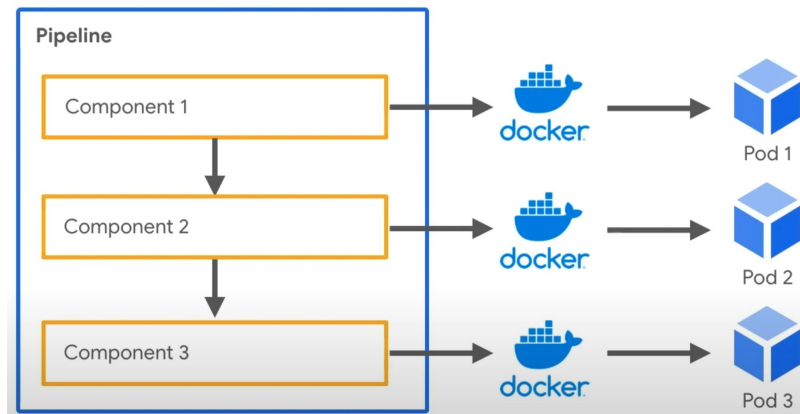
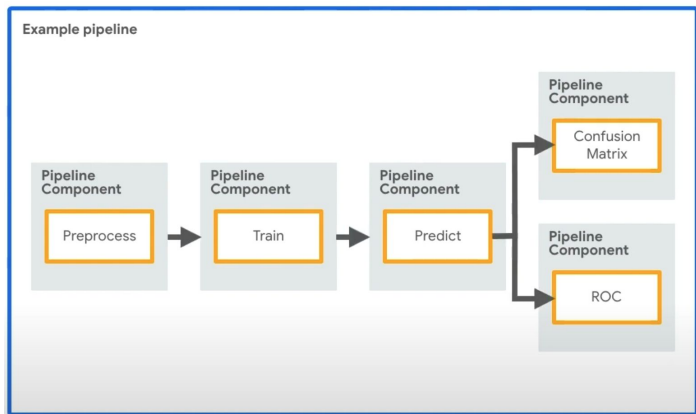
Machine Learning Pipelines

Automated ML workflows

A **user interface** (UI) for managing and tracking experiments, jobs, and runs

An **engine** for scheduling multi-step ML workflows

An **SDK/API** for defining and compiling pipelines and components



Benefits of Machine Learning Pipelines

Clear isolation between components

Can be scheduled to run periodically

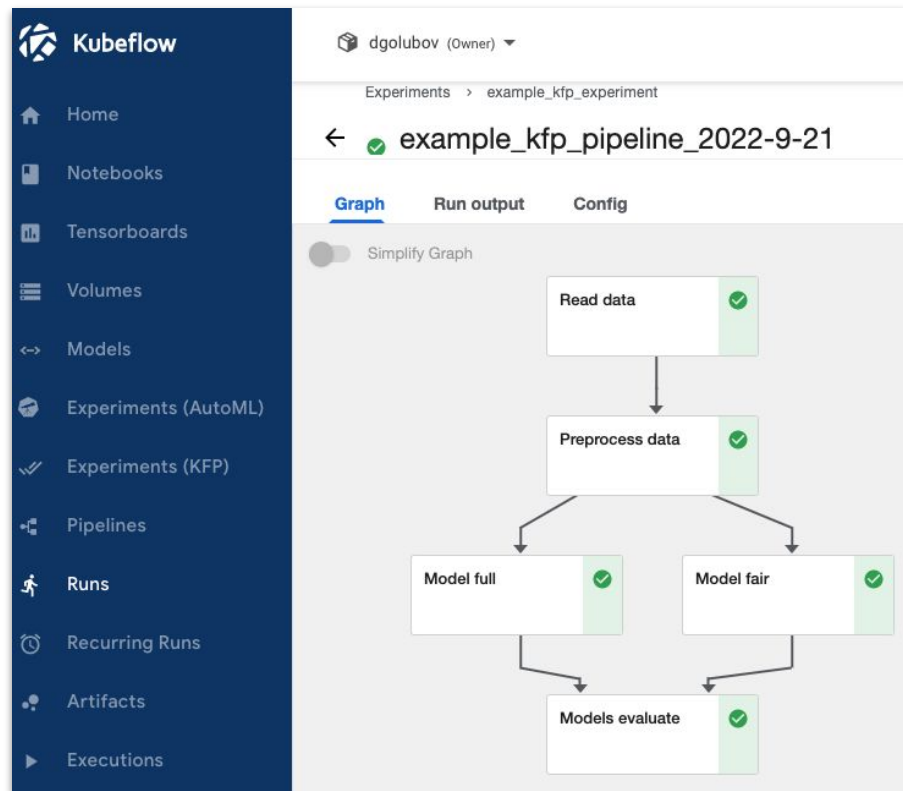
Can run with different input parameters

Versioning

Parallelisation

Non-blocking GPU access

Remote submissions with a client SDK



AutoML (Katib) - Hyperparameter Optimization



Standardized **development process**

Create a training script

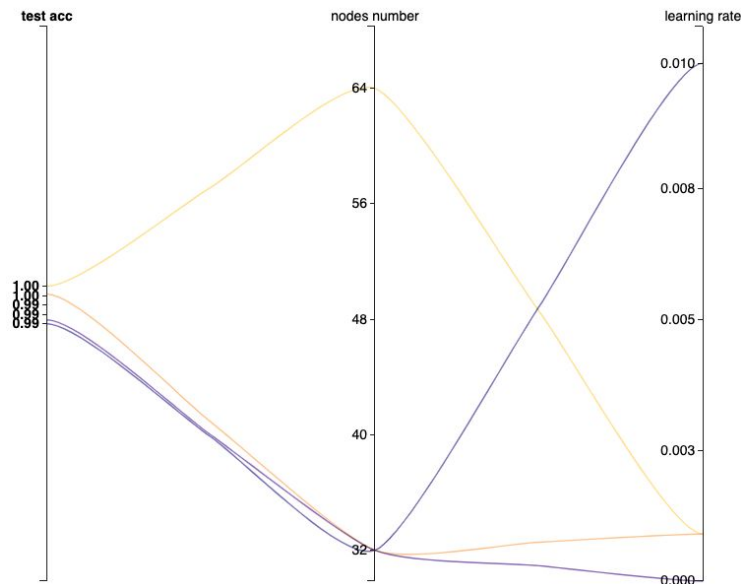
Build a Docker image

Run with various sets of inputs

Improved **hardware efficiency**

Run each trial on a separate GPU

Visualization of **results and metrics**



OVERVIEW	TRIALS	DETAILS	YAML	
Trial name	Status	Test acc	Nodes number	Learning rate
test-wze6q-bmmwdbc	Succeeded	0.99593	32	0.001
test-wze6q-kqvz9zcp	Succeeded	0.98831	32	0.01
test-wze6q-lnbhtzsz	Succeeded	0.98932	32	0.0001
test-wze6q-qvzh6pm8	Succeeded	0.99796	64	0.001

Distributed Training

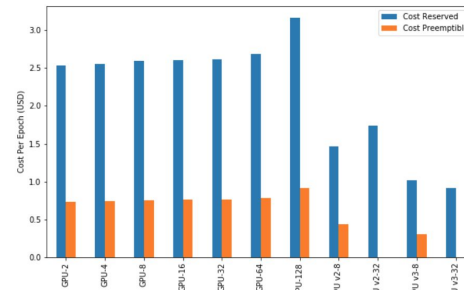
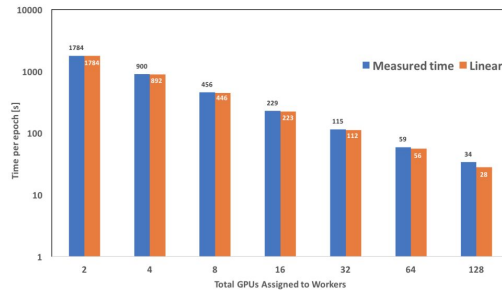
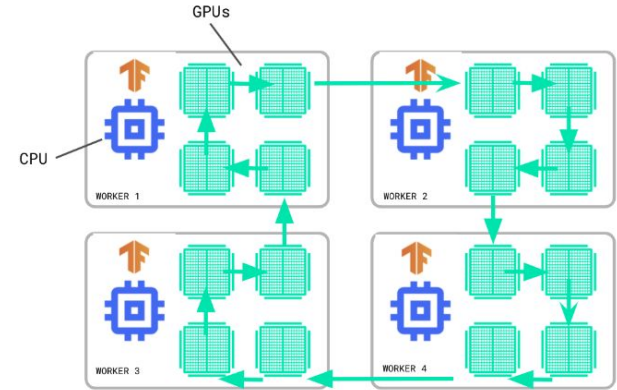
Major ML frameworks support distributed training

Training jobs split across multiple **local GPUs**

Kubeflow offers distributed training in Kubernetes

TFJob, PytorchJob, MXNetJob, MPIJob, XGBoostJob

Jobs split across multiple **cluster GPUs**



Accelerating GAN training using highly parallel hardware on public cloud, CHEP 2021

<https://doi.org/10.1051/epjconf/202125102073>

Tensorboards

Measurements and visualizations for ML workloads

Track **loss and accuracy**

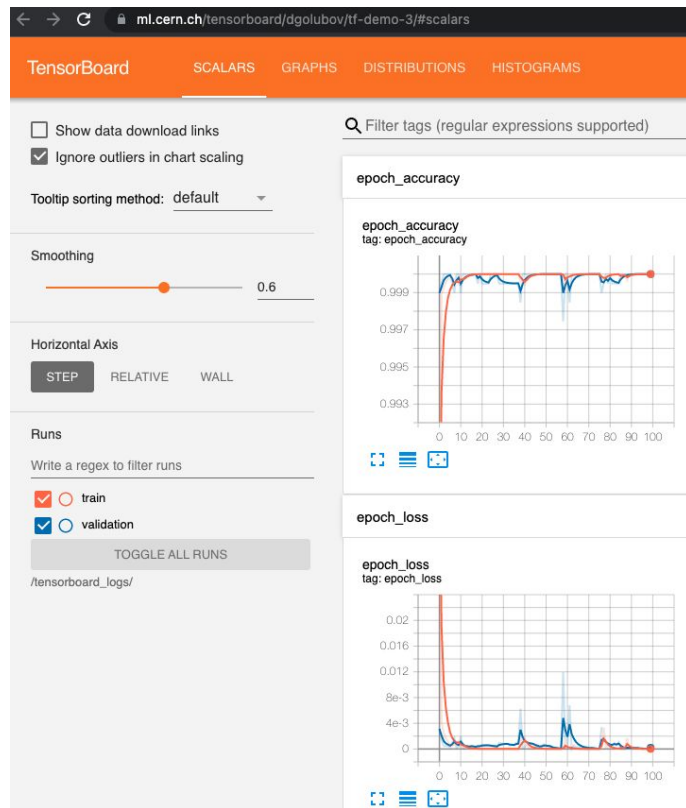
Visualize **model graph**

View custom metrics

Kubeflow allows creation of **Tensorboard servers**

Monitor model training real-time

Training from any Kubeflow component



Model Serving



Deploy a server to **run inference via http requests**

```
curl -v -H "Host: host" "http://host_ip/v1/models/mnist:predict" -d @./input.json
```

Serverless architecture

Automatic scaling per number of requests

Provided via **KServe** component



Demo

Resource Management

Resources assigned **per profile**

Memory, CPU, GPU, Kubernetes resources...

ResourceQuota defined for each profile

Personal profiles have a quota of **1 GPU** by default

Quotas for **group profiles** can be increased

For now by contacting us directly

Soon via dedicated ServiceNow form

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: kf-resource-quota
  namespace: dgolubov
status:
  hard:
    limits.cpu: "5"
    limits.memory: 10Gi
    limits.nvidia.com/gpu: "1"
    (soon) limits.nvidia.com/gpu.shared: "1"
    requests.cpu: "5"
    requests.memory: 10Gi
    requests.nvidia.com/gpu: "1"
    (soon) requests.nvidia.com/gpu.shared: "1"
  persistentvolumeclaims: 15
  pods: 100
```


Resource Management - Upcoming

Support for **time-sliced** NVIDIA GPUs

Multiple pods can access a single GPU, allowing for time sharing

Useful for smaller workloads, ex. notebooks with infrequent GPU utilization

Support for **physically sliced** NVIDIA GPUs

A GPU memory is physically split, without concurrent access

Useful for medium to large workloads that require constant GPU access

A ServiceNow form for requesting resources for group profiles

Storage Integration

EOS supported with Kerberos authentication

CVMFS via CSI

PVCs with Openstack Manila

S3 object storage supported via S3 clients authentication

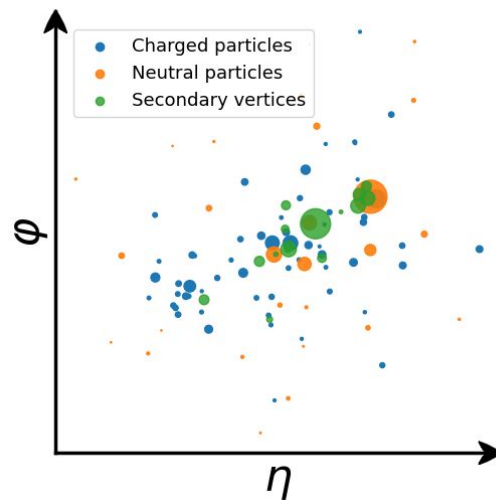
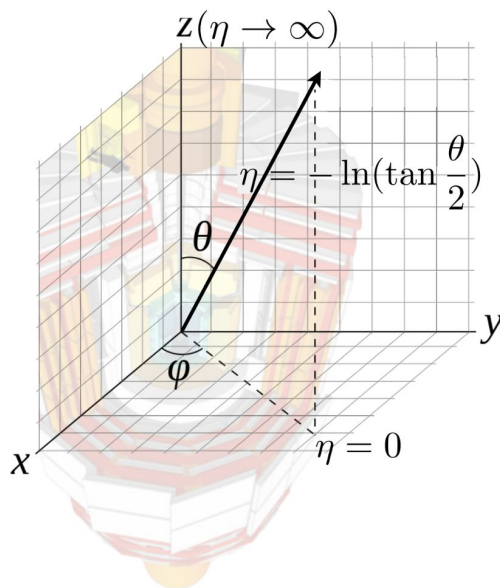
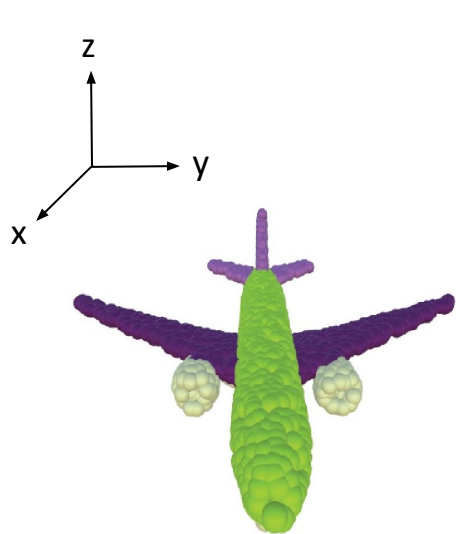
s3.cern.ch or public cloud providers (Amazon S3, Google Object Storage...)

registry.cern.ch - registry for the built images

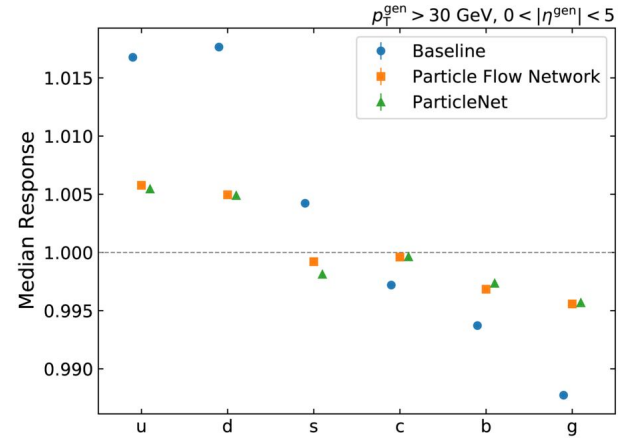
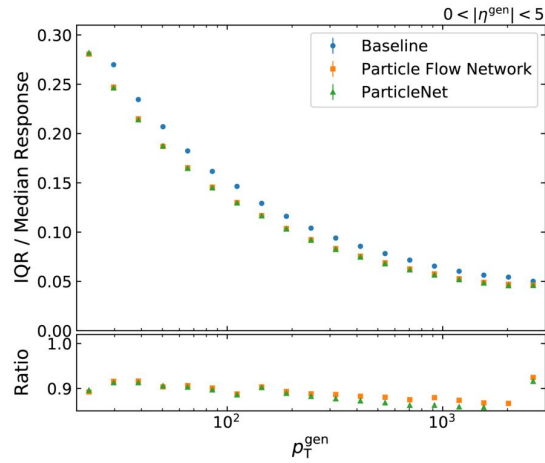
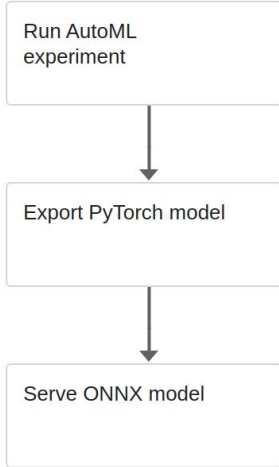
Use Case - CMS Jet Tagging

Use detector coordinates to represent jets as **particle clouds**

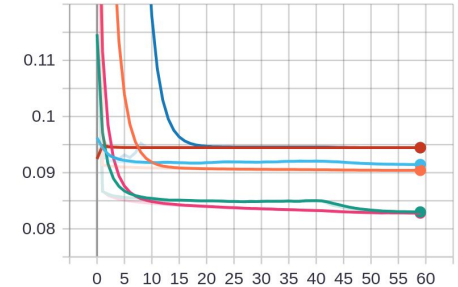
Analogous to **point clouds** in computer vision problems



Use Case - CMS Jet Tagging



train_epoch_
tag: Loss/train_epoch_



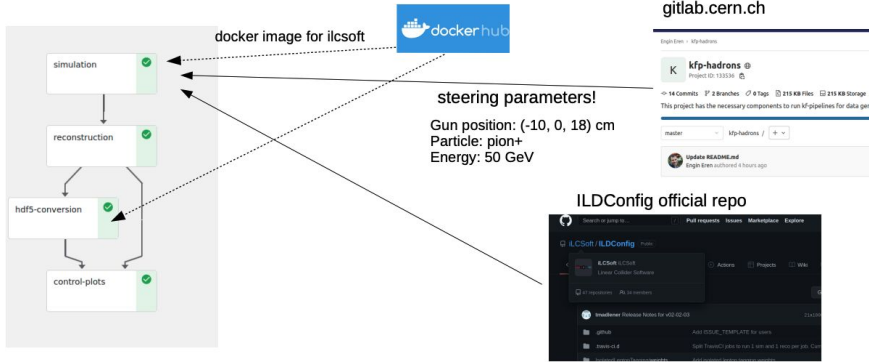
Model Servers

[+ NEW MODEL SERVER](#)

Status	Name	Age	Predictor	Runtime	Protocol	Storage URI		
✓	particle-net-regressor-...	1 month ago	Triton	21.09-py3		s3://jec-data/particle-net-regressor-25a03c		
✓	pfn-regressor-ea37f4	2 months ago	Triton	21.09-py3		s3://jec-data/pfn-regressor-ea37f4		

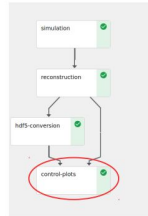
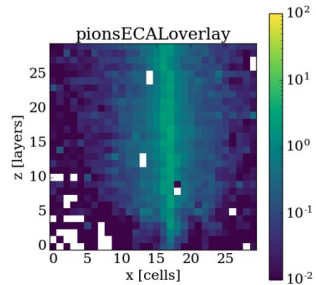
Use Case - 3DGAN Desy

Data pipeline and experiment

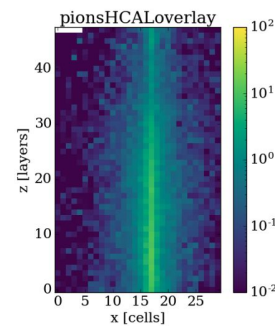


Writing all hdf5 files to CERN-EOS

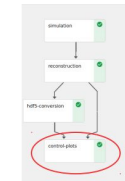
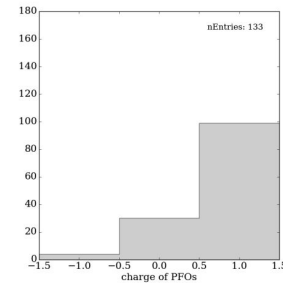
ECAL (z-x)



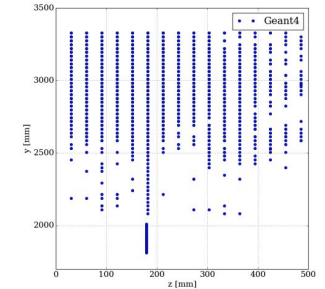
HCAL (z-x)



Pandora PFO:



Simulated hits:



Other Use Cases

OpenLab 3DGAN - AutoML, Distributed Tensorflow Training

UNOSAT - Pytorch Training, Pipelines

ATLAS Spanet - Pipelines, AutoML, Inference

ADMON Anomaly Detection - Inference Services

Conclusions

Community effort to improve machine learning infrastructure

Kubeflow ongoing active development

CERN users can influence future developments

Anyone can contribute to open source

<https://github.com/kubeflow>

Everyone is invited to **provide feedback!**

Previous Talks

KubeCon Europe 2022, May 17 2022

[Jet Energy Corrections with GNN Regression using Kubeflow @ CERN](#)

KubeCon North America 2021, October 12 2021

[A Better and More Efficient ML Experience for CERN Users](#)

KubeCon Europe 2021, May 6 2021

[Building and Managing a Centralized ML Platform with Kubeflow at CERN](#)

25th International Conference on Computing in High-Energy and Nuclear Physics, May 20 2021

[Training and Serving ML workloads with Kubeflow at CERN](#)

Fast Machine Learning for Science Workshop, Dec 01 2020

[Making ML Easier with Kubeflow](#)

Links

<https://ml.cern.ch> - the service landing page

<ml.docs.cern.ch> - documentation pages

<https://gitlab.cern.ch/ai-ml/examples> - examples repository

<https://mattermost.web.cern.ch/it-dep/channels/ml> - Mattermost channel

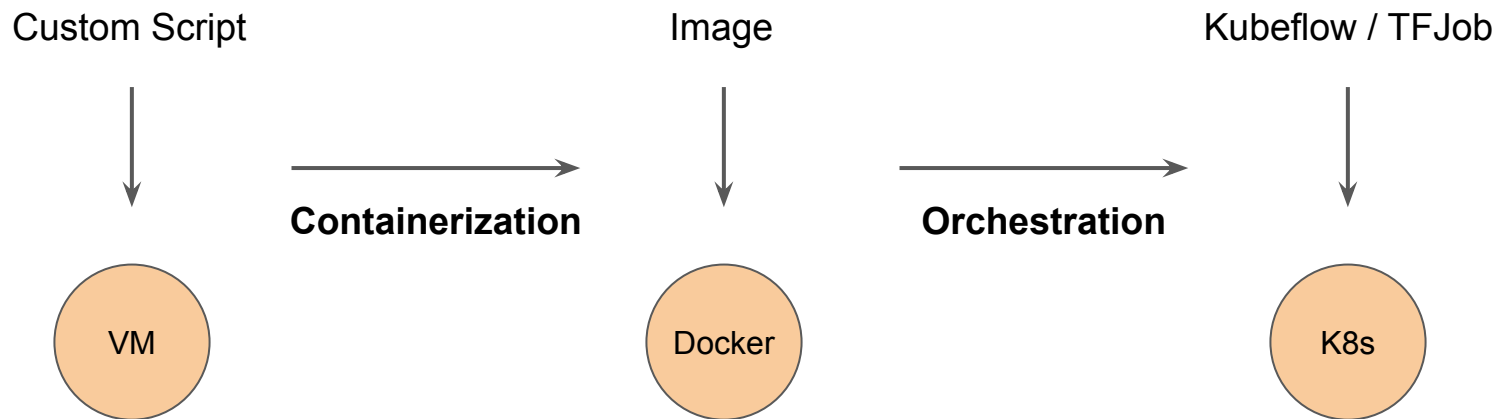
Others may benefit from your questions!

Thank you!

Q&A

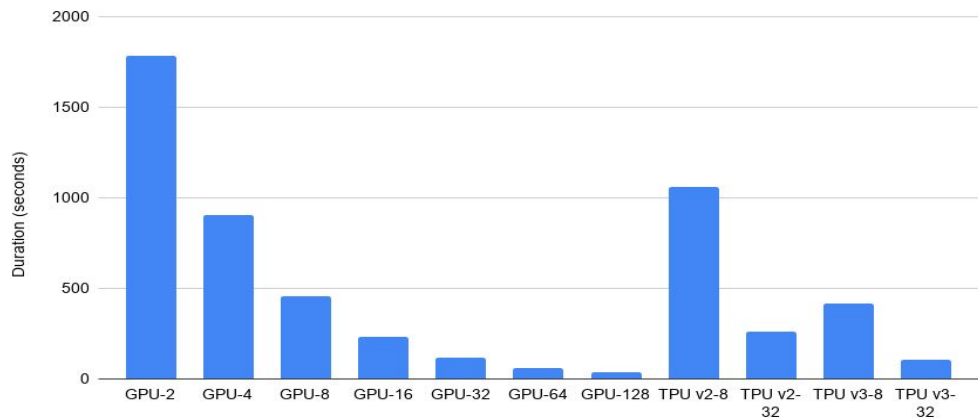
A Cloud Native Showcase

From a custom script to a large distributed training...



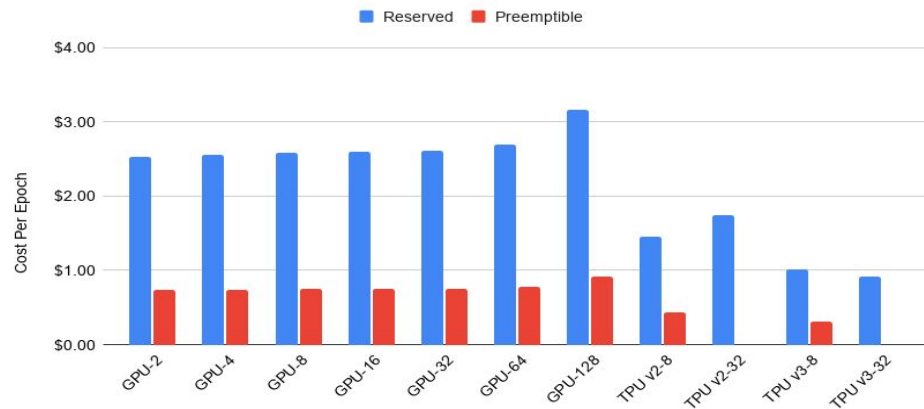
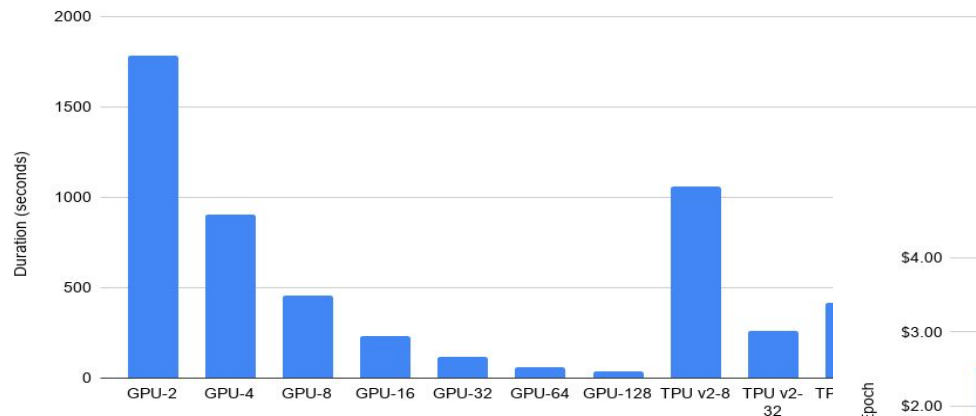
A Cloud Native Showcase

From a custom script to a large distributed training...



A Cloud Native Showcase

From a custom script to a large distributed training...



A Cloud Native Showcase

From a custom script to a large distributed training...

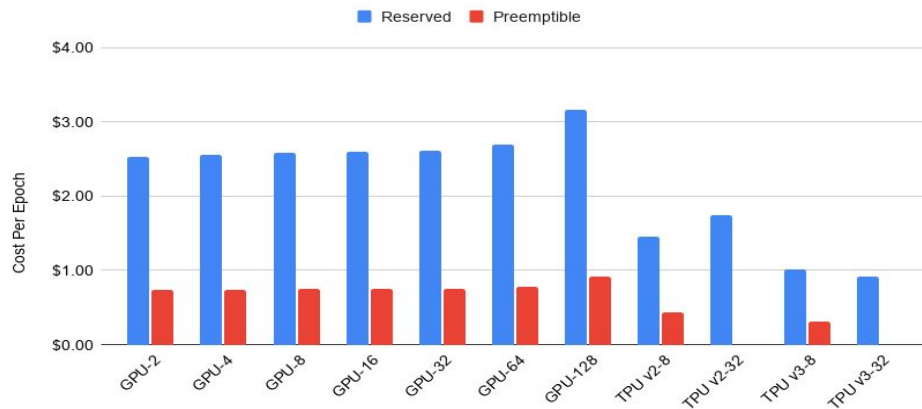
1 to 128 GPUs: 3550 to 35 seconds per epoch

x100 speedup

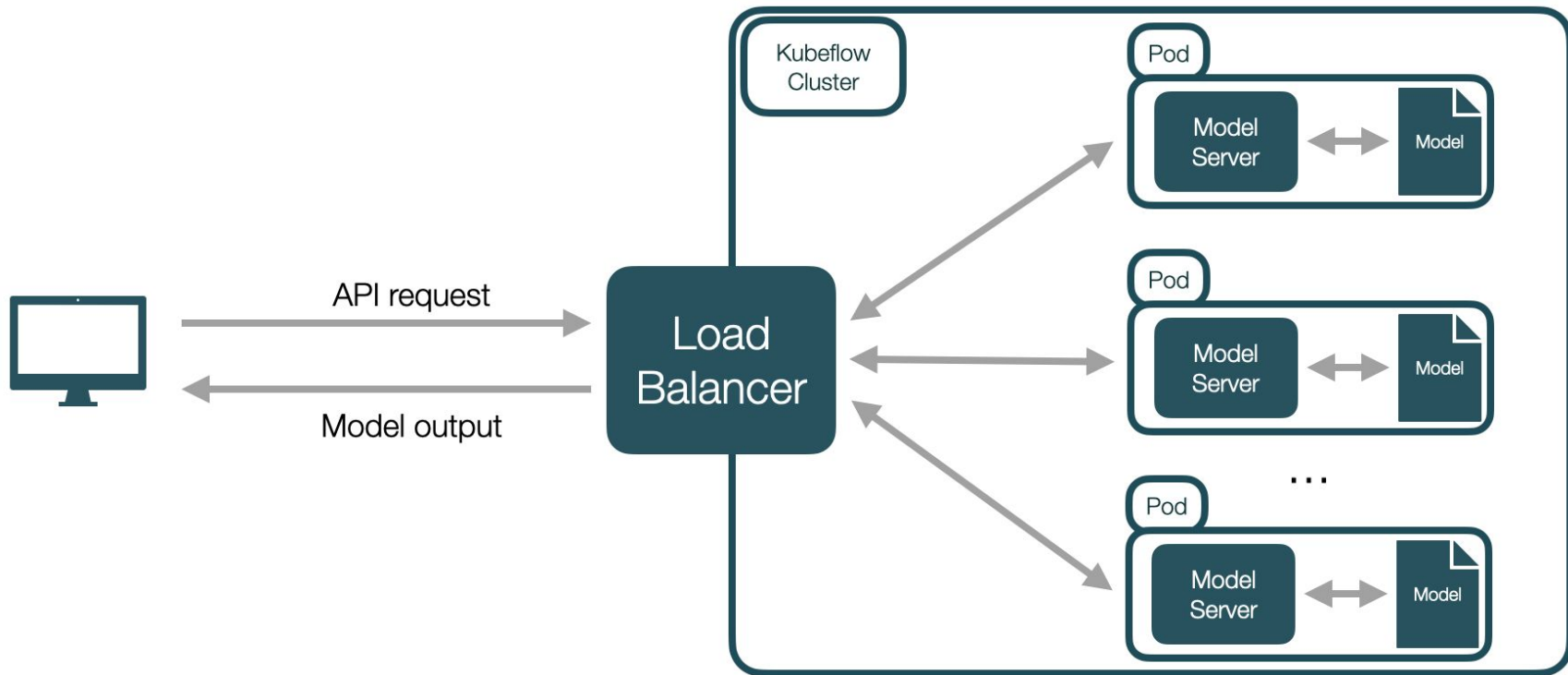
Almost the same total cost

How to handle spot?

How to integrate multiple regions?



Model Serving



Current GPU Usage

Overall utilization ~30%

Largely limited by idle notebooks *holding* GPUs

Solution out soon

Notebooks: time sliced

Pipelines: full cards/partitions

