CERN openlab

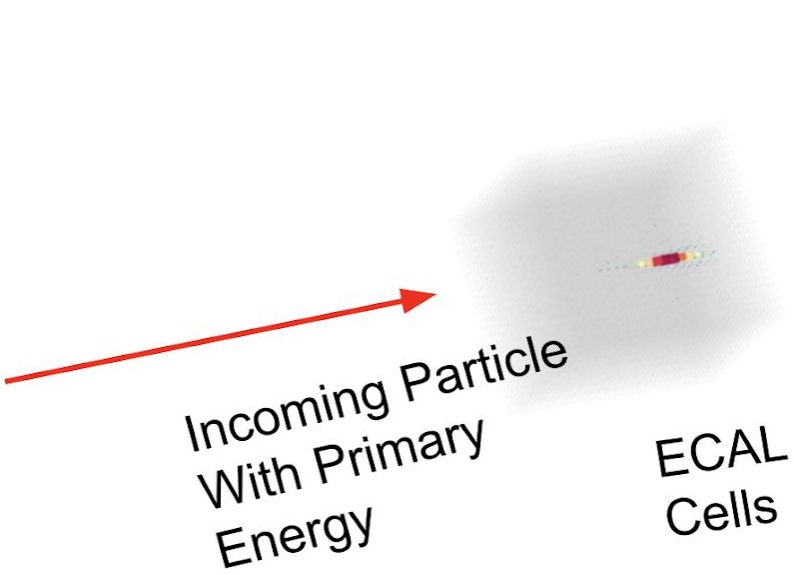
Machine Learning and Kubernetes

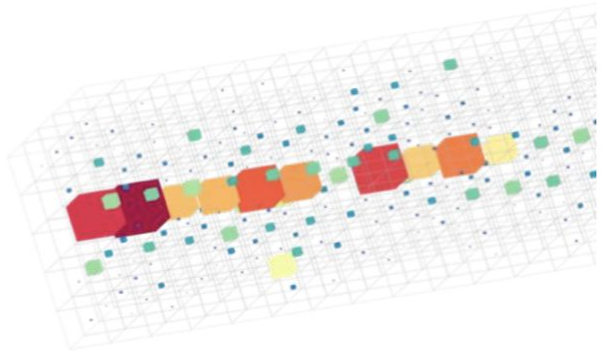
Fedor Kitashov
Supervisor: Ricardo Rocha

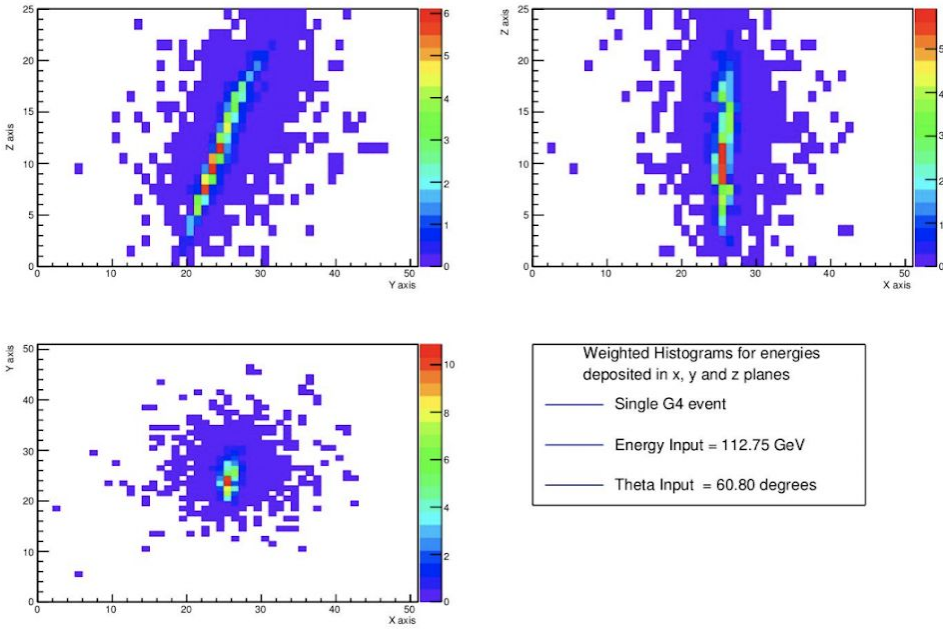
Plan

- Simulations using GANs
- Why we need GPUs
- Docker and Kubernetes

CLIC Calorimeter simulation



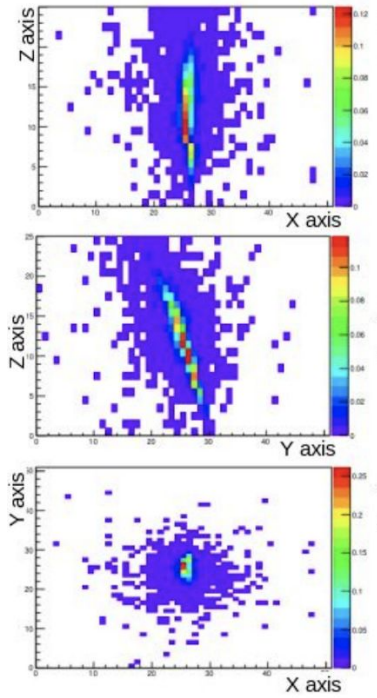




Source: https://indico.cern.ch/event/766872/contributions/3357987/attachments/1831267/2999045/iml_2019_3dgan.pdf

G4 vs 3D GAN

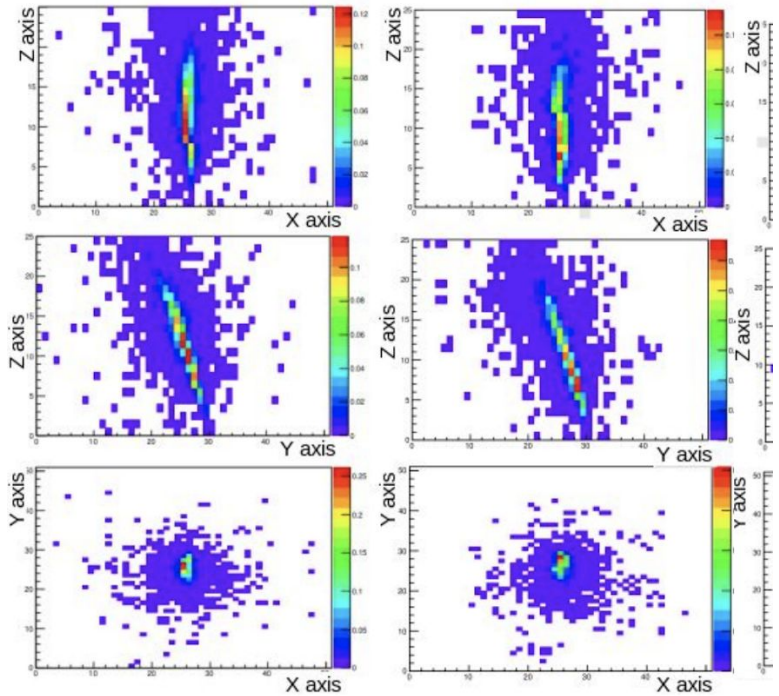
111.07 GeV and 115.54 Degrees



G4

G4 vs 3D GAN

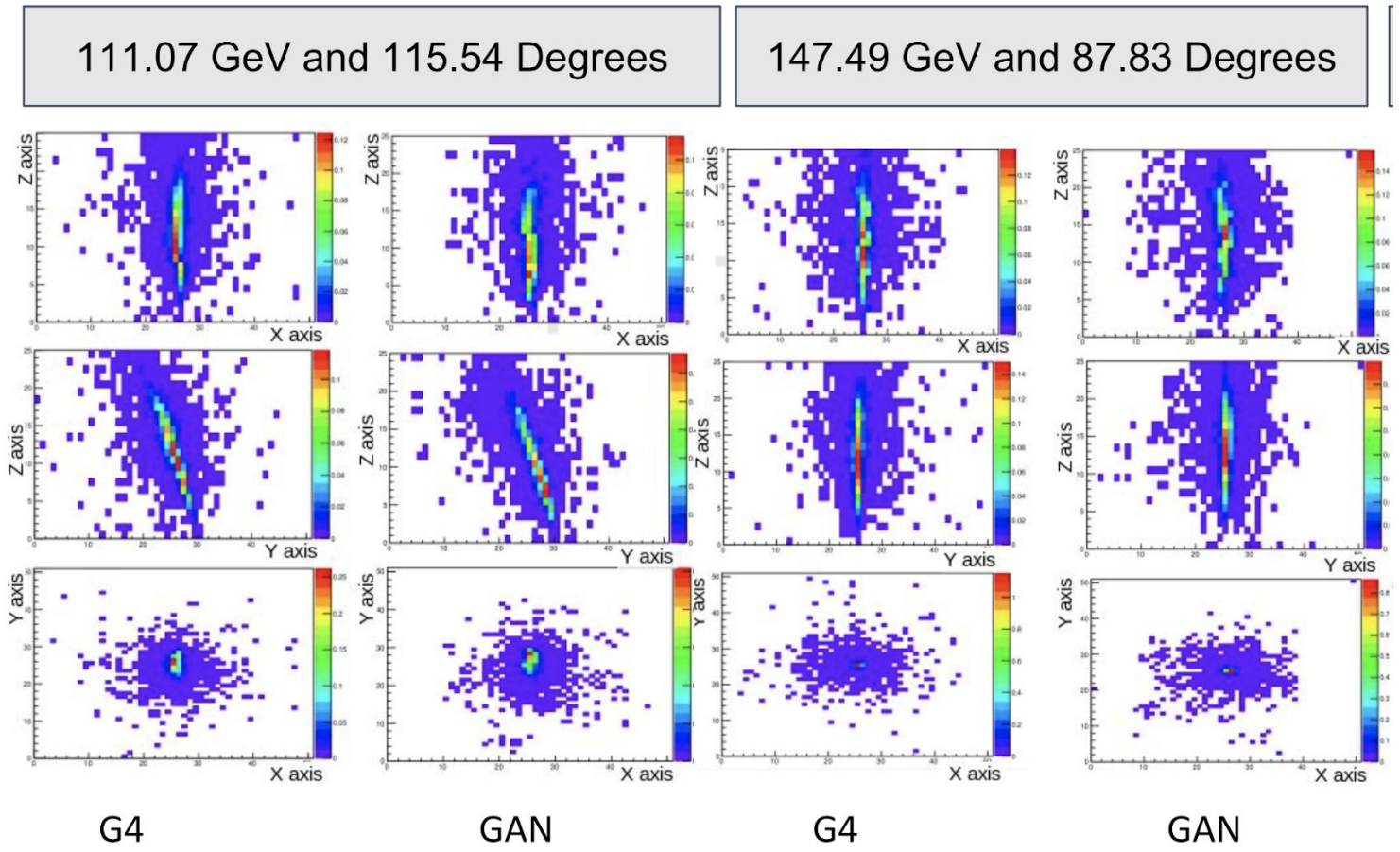
111.07 GeV and 115.54 Degrees



G4 GAN

Source: https://indico.cern.ch/event/766872/contributions/3357987/attachments/1831267/2999045/iml_2019_3dgan.pdf

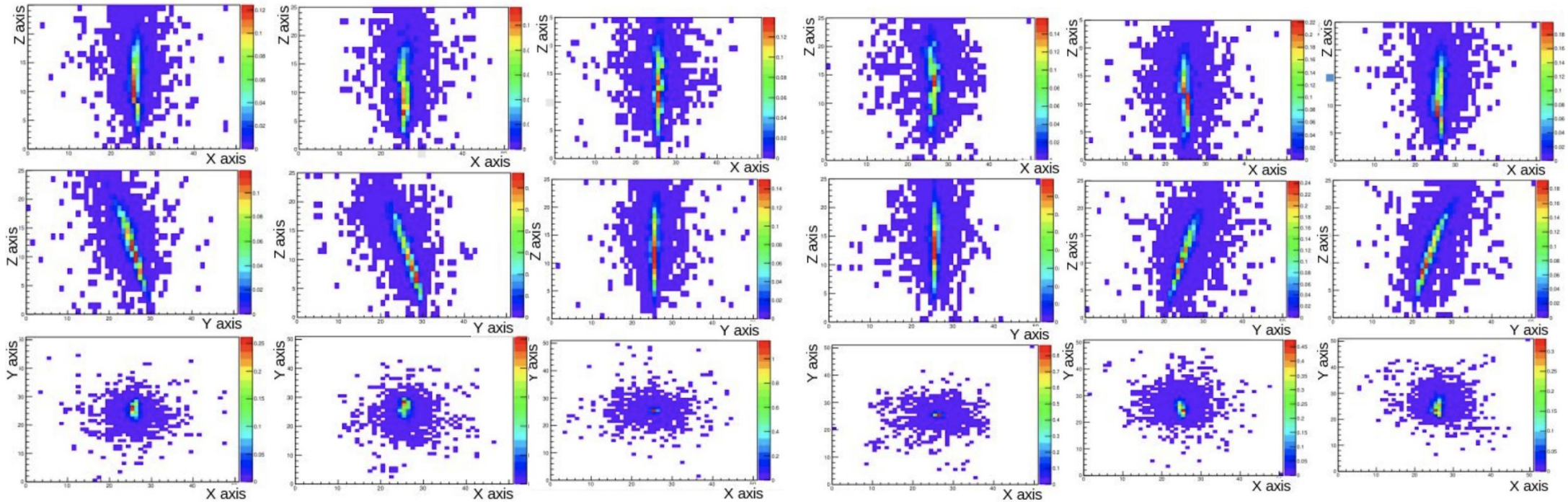
G4 vs 3D GAN



Source: https://indico.cern.ch/event/766872/contributions/3357987/attachments/1831267/2999045/iml_2019_3dgan.pdf

G4 vs 3D GAN

111.07 GeV and 115.54 Degrees		147.49 GeV and 87.83 Degrees		188.95 GeV and 62.96 Degrees	
G4	GAN	G4	GAN	G4	GAN



Source: https://indico.cern.ch/event/766872/contributions/3357987/attachments/1831267/2999045/iml_2019_3dgan.pdf

Timing

Fixed Angle

- Fixed Angle:
  - Inference is four order of magnitude faster
  - Training time < 1 hour GTX 1080
- Variable Angle:
  - Training time 2-3 hours GTX 1080

Time to create an electron shower for Fixed angle		
Method	**Machine**	**Time/Shower (msec)**
Full Simulation Fixed Energy (Geant 4)	Intel Xeon Platinum 8180	17000
3DGAN (Fixed Angle) (batch size 128)	Intel Xeon Platinum 8180(TF 1.12)	1
	GTX 1080	0.1
3DGAN (Variable Angle) (batch size 64)	GTX 1080	4.6

Source: https://indico.cern.ch/event/766872/contributions/3357987/attachments/1831267/2999045/iml_2019_3dgan.pdf

Dependencies are painful sometimes

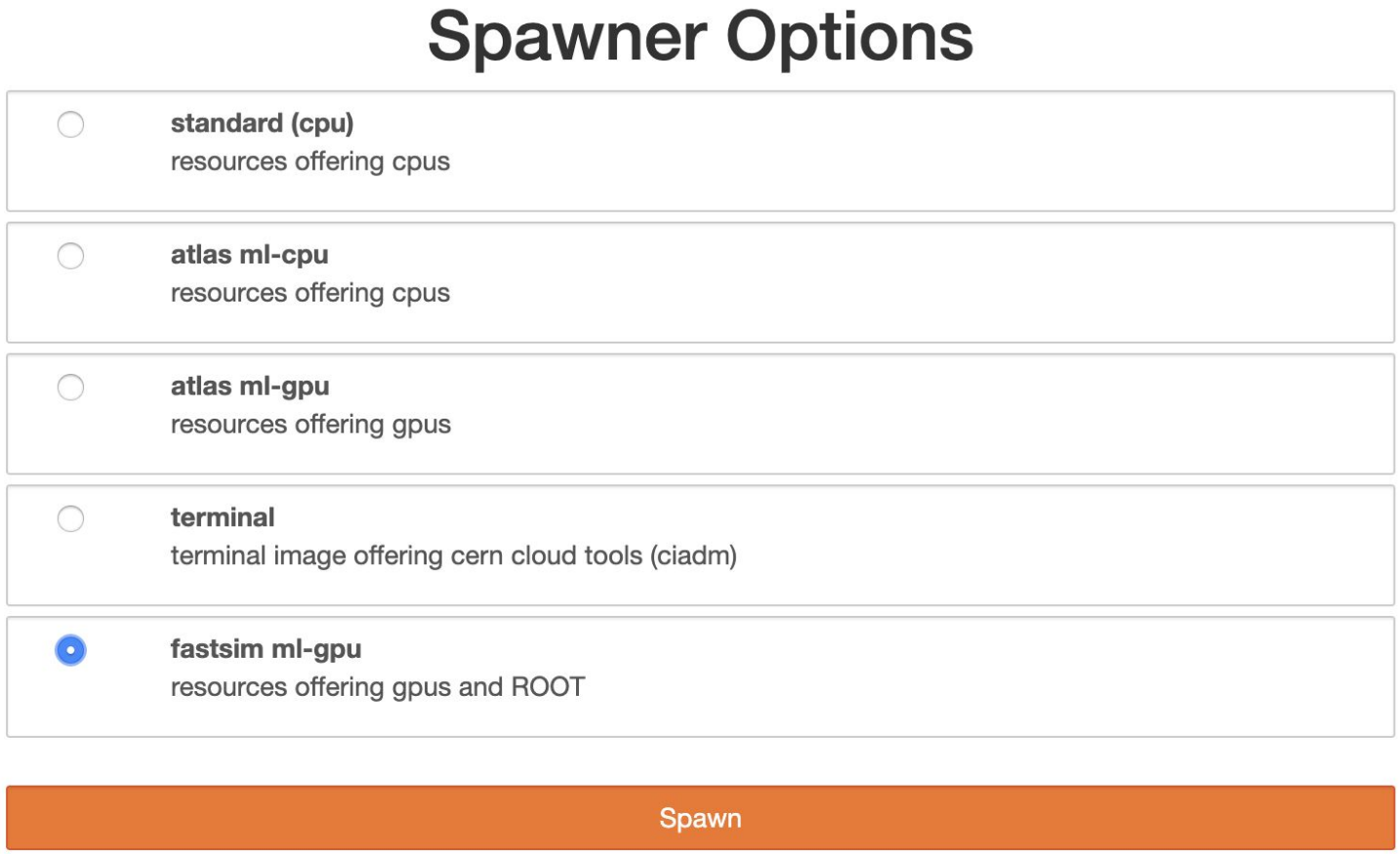
```
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcudnn.so locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcufft.so locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcuda.so.1 locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcurand.so locally
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
modprobe: ERROR: could not insert 'nvidia_361_uvm': Invalid argument
E tensorflow/stream_executor/cuda/cuda_driver.cc:491] failed call to cuInit: CUDA_ERROR_UNKNOWN
I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:153] retrieving CUDA diagnostic information for host: pascal
I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:160] hostname: pascal
I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:185] libcuda reported version is: 361.93.2
I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:356] driver version file contents: """NVRM version: NVIDIA UNIX x86_64 Kernel Module  352.99  Mon Jul  4 23:52:14 PDT 2016
GCC version:  gcc version 4.8.4 (Ubuntu 4.8.4-2ubuntu1~14.04.3)
"""
I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:189] kernel reported version is: 352.99.0
E tensorflow/stream_executor/cuda/cuda_diagnostics.cc:296] kernel version 352.99.0 does not match DSO version 361.93.2 -- cannot find working devices in this configuration
I tensorflow/core/common_runtime/gpu/gpu_init.cc:81] No GPU devices available on machine.

 grep -r nvidia /etc/modprobe.d/
/etc/modprobe.d/blacklist-framebuffer.conf:blacklist nvidiafb
/etc/modprobe.d/fbdev-blacklist.conf:blacklist nvidiafb
/etc/modprobe.d/nvidia-361_hybrid.conf:# This file was installed by nvidia-361
/etc/modprobe.d/nvidia-352_hybrid.conf:# This file was installed by nvidia-352
```

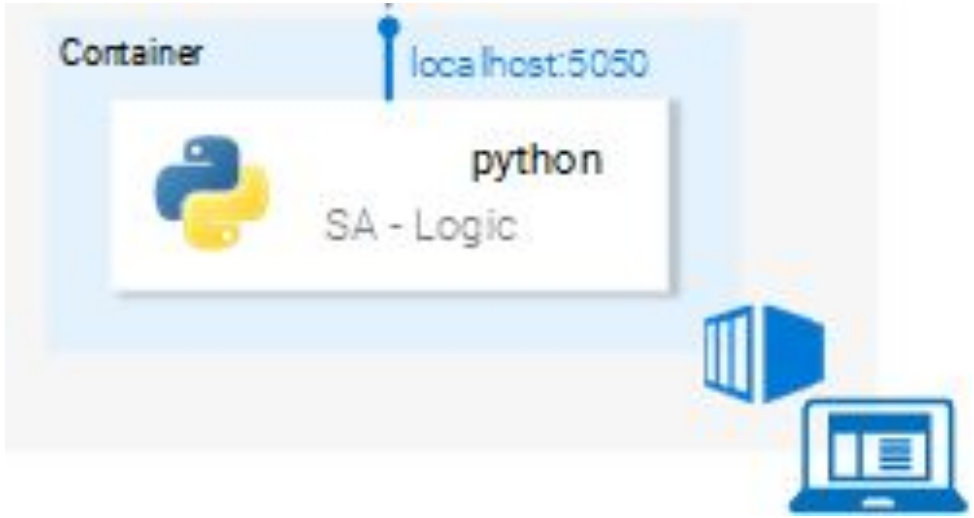
What is Docker and why you need it

```
(base) kitashov@kitashov:~$ docker run -it -p 8889:8889 gitlab-registry.cern.ch/cloud/fastsim
```

JupyterHub

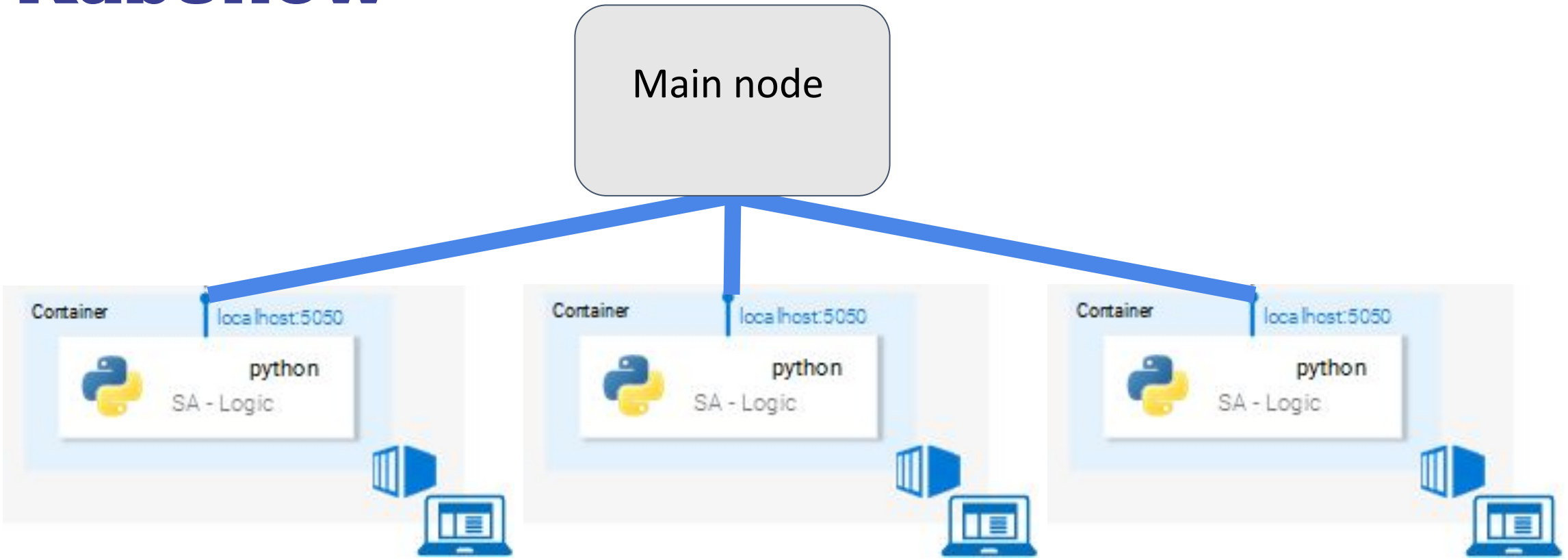

Spawner Options
standard (cpu)
resources offering cpus
atlas ml-cpu
resources offering cpus
atlas ml-gpu
resources offering gpus
terminal
terminal image offering cern cloud tools (ciadm)
fastsim ml-gpu
resources offering gpus and ROOT
Spawn

Dockerized application



Source:
https://www.freecodecamp.org/news/learn-kubernetes-in-under-3-hours-a-detailed-guide-to-orchestrating-containers-114ff420e882/

Kubeflow



Source:
https://www.freecodecamp.org/news/learn-kubernetes-in-under-3-hours-a-detailed-guide-to-orchestrating-containers-114ff420e882/

Contribution and Future Work

- We created a reliable Docker image with ROOT and GPU-powered Tensorflow
- We tested simulations on a single GPU
- We are working towards using Kubeflow to make experiments more stable and scalable

Thank you!

fedor.kitashov@phystech.edu