MLaaS4HEP: Machine Learning as a Service for HEP

Luca Giommi

University of Bologna and INFN Bologna, Italy

luca.giommi@cern.ch

tCSC security 2022 - Split, Croatia

MLaaS for HEP

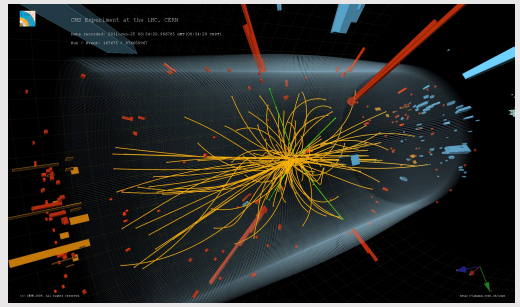
Machine Learning (**ML**) techniques in the High Energy Physics (**HEP**) domain are ubiquitous, successfully used in many areas, and will play a significant role also in Run3 and High-Luminosity LHC upgrade.

It is necessary to have a synergy between HEP and ML community and to ease the usage of ML techniques in HEP analyses. Therefore, it would be useful to provide physicists who are not experts in ML a service to easily exploit the ML potentiality.

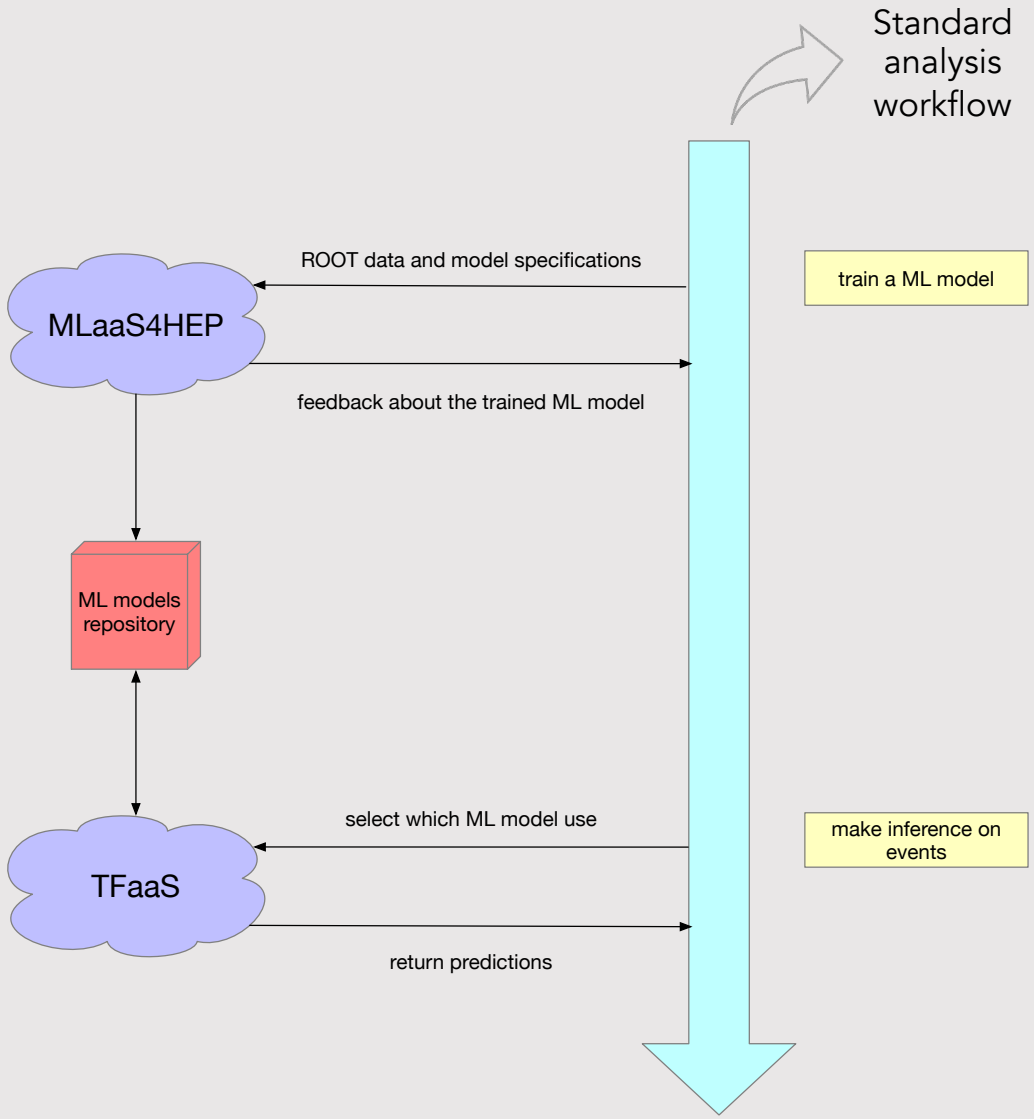
Existing ML as a Service (**MLaaS**) solutions are many: they offer many services and cover different use cases but they are not directly usable in HEP.

Existing HEP R&D solutions don't cover the whole ML pipeline or they are not "aaS" solutions or they are difficult to generalize to other use cases.

We proposed a MLaaS for HEP (**MLaaS4HEP**) solution as a product of R&D activities within the CMS experiment.



Basic idea of a ML as a Service for HEP workflow



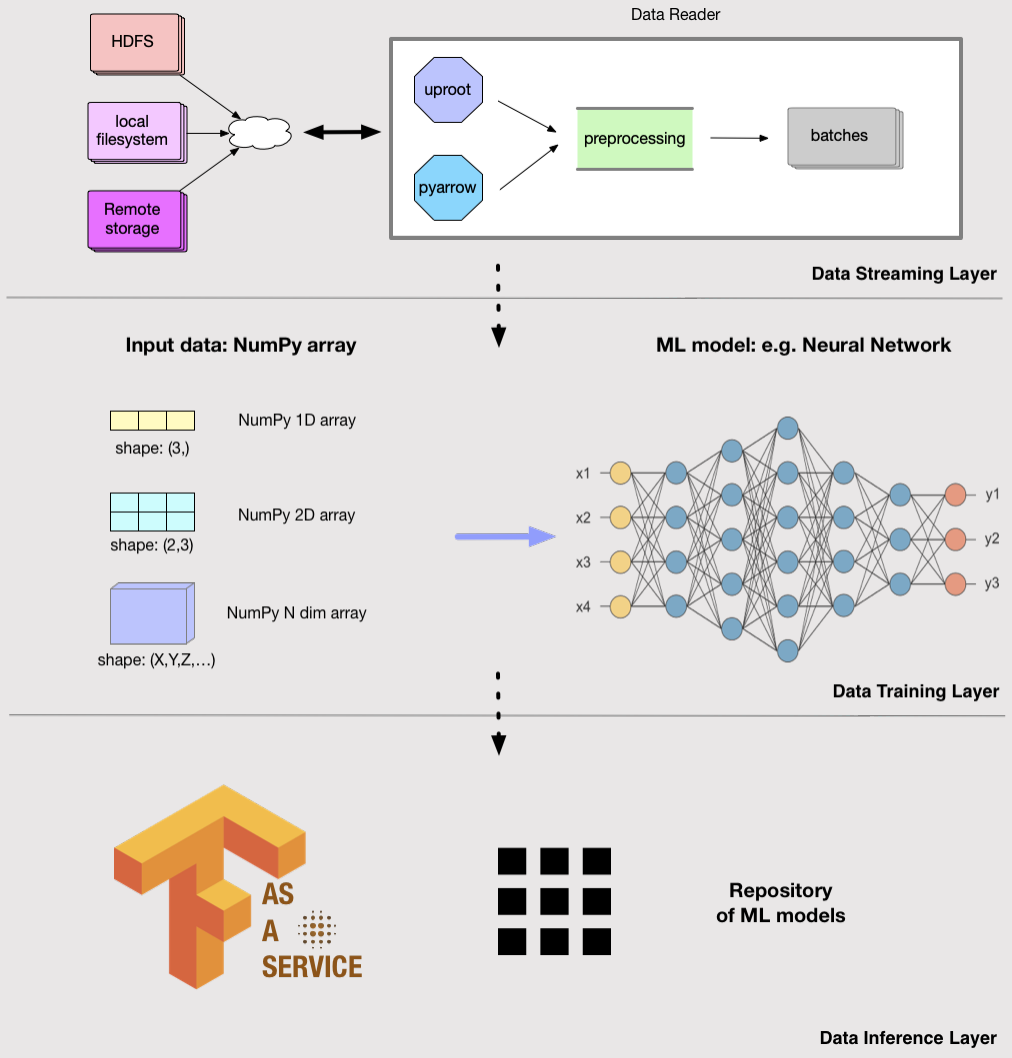
MLaaS4HEP project

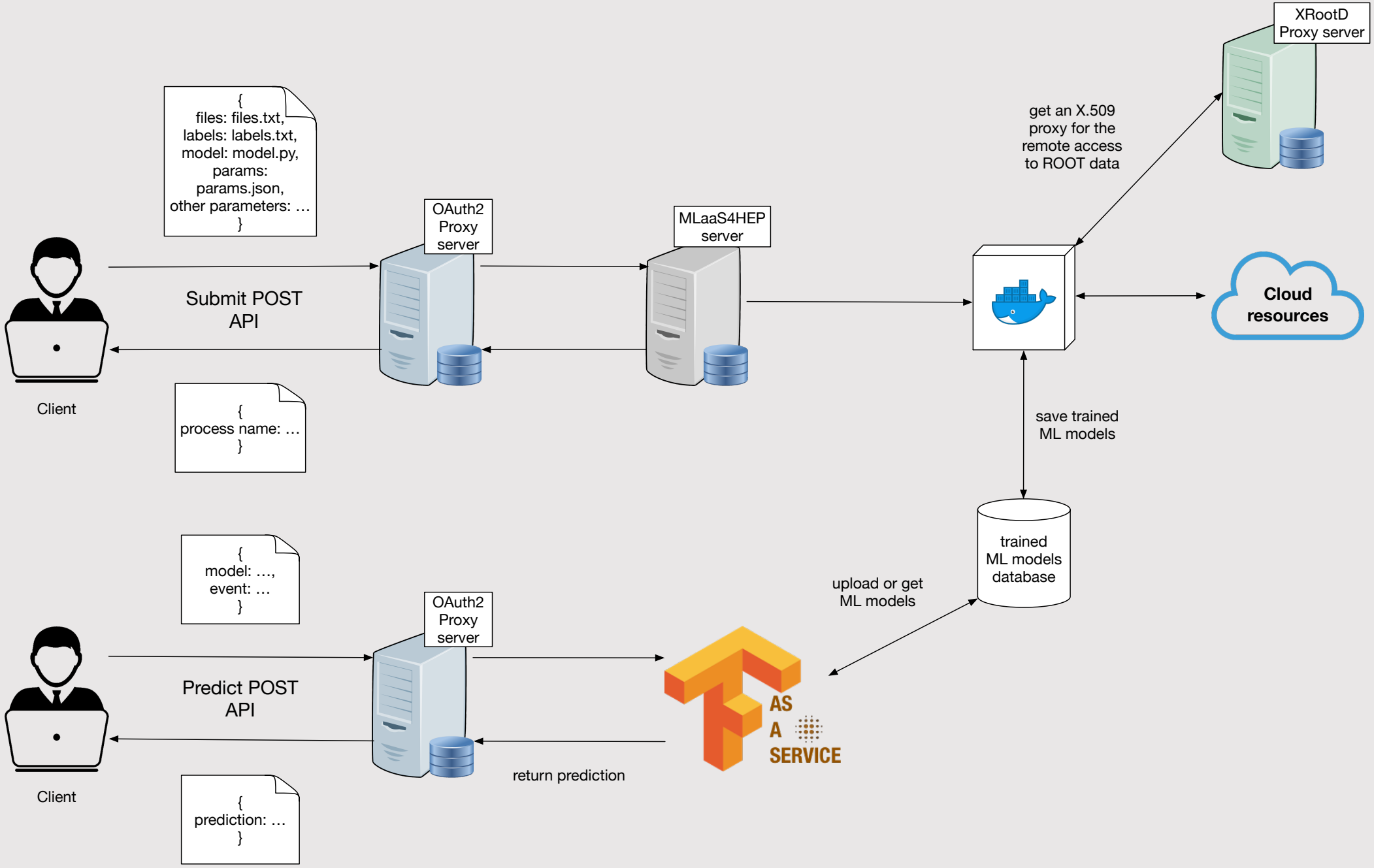
The MLaaS4HEP framework has a multi-language architecture (Python and Go) and it is structured in three layers:

- **Data Streaming Layer** is responsible for local and remote data access of HEP ROOT files (based on the uproot library)
- **Data Training Layer** is responsible for feeding HEP ROOT data into the ML framework of user choice
- **Data Inference Layer** (TFaaS) provides access to pre-trained TF models via HTTP protocol

The first two layers contributes to the **MLaaS4HEP** training workflow, while **TFaaS** is encharged of the inference phase.

Data streaming and training tools: github.com/vkuznet/MLaaS4HEP
Data inference tool: github.com/vkuznet/TFaaS




{
files: files.txt,
labels: labels.txt,
model: model.py,
params:
params.json,
other parameters: …
}
OAuth2
Proxy
server
MLaaS4HEP
server
XRootD
Proxy server
get an X.509
proxy for the
remote access
to ROOT data
Cloud
resources
Submit POST
API
Client
{
process name: …
}
save trained
ML models
trained
ML models
database
{
model: …,
event: …
}
upload or get
ML models
OAuth2
Proxy
server
AS
A
SERVICE
Predict POST
API
Client
return prediction
{
prediction: …
}

Towards MLaaS4HEP as cloud service

The goal is to create a **cloud service** that could use cloud resources and could be added into the INFN Cloud portfolio of services. To provide this we worked through a series of steps.

- We built a **MLaaS4HEP server** with some basic APIs using the (Python-based) Flask framework
- We integrated an **OAuth2 Proxy server** to manage users authentication/authorization
- We integrated an **XRootD Proxy server** to allow the usage of an X.509 proxy for the remote access to ROOT data
- We connected the MLaaS4HEP server and **TFaaS** in a way that the ML models trained by the MLaaS4HEP server are saved in a repository from which the TFaaS service can take them for the inference phase

We implemented a working prototype connecting the aforementioned services, hosted by a VM of the INFN Cloud. The MLaaS4HEP server APIs can be reached at the following address **https://90.147.174.27:4433** while TFaaS at **https://90.147.174.27:8081**.

DEMO

MLaaS4HEP project

- V. Kuznetsov, L. Giommi, D. Bonacorsi, *MLaaS4HEP: Machine Learning as a Service for HEP*. Comput Softw Big Sci 5, 17. DOI: 10.1007/s41781-021-00061-3
- L. Giommi, D. Spiga, V. Kuznetsov, D. Bonacorsi, *Prototype of a cloud native solution of Machine Learning as Service for HEP*. Soon it will be submitted to the ICHEP proceedings
- L. Giommi, D. Spiga, V. Kuznetsov, D. Bonacorsi, M. Paladino, *Cloud native approach for Machine Learning as a Service for High Energy Physics*. Submitted to the proceedings (PoS ISGC2022, 012)
- L. Giommi, V. Kuznetsov, D. Bonacorsi, D. Spiga, *Machine Learning as a Service for High Energy Physics on heterogeneous computing resources*. PoS ISGC2021, 019. DOI: 10.22323/1.378.0019
- L. Giommi, D. Bonacorsi, V. Kuznetsov, *Prototype of Machine Learning "as a Service" for CMS Physics in Signal vs Background discrimination*. PoS LHCP2018, 093. DOI: 10.22323/1.321.0093

Thanks for the attention

Backup slides

```
./workflow.py --files=files.txt --labels=labels.txt --model=model.py --params=params.json
```

MLaaS workflow | Input ROOT files | Labels of ROOT files | User model | MLaaS parameters

Keras model (model.py)

```
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout

def model(idim):
    "Simple Keras model for testing purposes"
    ml_model = Sequential([Dense(128,
activation='relu',input_shape=(idim,)),
                           Dropout(0.5),
                           Dense(64, activation='relu'),
                           Dropout(0.5),
                           Dense(1, activation='sigmoid')])
    ml_model.compile(optimizer=keras.optimizers.Adam(lr=1e-3),
                     loss=keras.losses.BinaryCrossentropy(),
                     keras.metrics.AUC(name='auc')])
```

MLaaS parameters (params.json)

```
{
    "nevts": 30000,
    "shuffle": true,
    "chunk_size": 10000,
    "epochs": 5,
    "batch_size": 100,
    "identifier":["runNo", "evtNo", "lumi"],
    "branch": "boostedAk8/events",
    "selected_branches":"",
    "exclude_branches": "",
    "hist": "pdfs",
    "redirector": "root://xrootd.ba.infn.it",
    "verbose": 1
  }
```

Input ROOT files (files.txt)

```
PATH/flatTree_ttHJetTobb_M125_13TeV_amcatnloFXFX_madspin_pythia8.root
PATH/flatTree_TT_TuneCUETP8M2T4_13TeV-powheg-pythia8.root
```

Labels of ROOT files (labels.txt)

```
1
0
```

MLaaS parameters

Read remote root files

Write and load the specs

```
./workflow.p  --files=files.txt --labels=  bels.  xt –model=model.py --params=params.json
DataGenerato . <MLaaS4HEP.generator.Roo  ataGen  ator object at 0x7f0cb58d7fd0> [29/Jun/2020:17:53:  4] 1593445994.0
model parameters: {"nevts": 30000, "s  ffle": t  ue, "chunk_size": 10000, "epochs": 2, "batch_size"  500, "identifier": ["runNo", "evtNo", "lumi"],
"branch": "boostedAk8/events", "sel  ted_branc  s": "", "exclude_branches": "", "hist": "pdfs", "r  director": "root://xrootd.ba.infn.it", "verbose": 1}


Reading root://xrootd.ba.infn.it//PATH_FILES/  latTree_ttHJetTobb_M125_13TeV_amcatnloFXFX_madspin  ythia8.root
# 10000 entries, 77 branches, 9.522203445434  7 MB, 1.0169336795806885 sec, 9.36364252323795 MB/s  c, 9.833482950553169 kHz
# 10000 entries, 77 branches, 9.533915519714  55 MB, 1.2977769374847412 sec, 7.346343770133804 M  sec, 7.705484441248654 kHz
# 10000 entries, 77 branches, 9.538667678833  08 MB, 1.4104814529418945 sec, 6.7627033726234735  B/sec, 7.089777734505208 kHz
--- first pass: 948348 events, (22-flat, 5  jagged) branches, 328 attrs
<MLaaS4HEP.reader.RootDataReader object at  0x7f840dbf4d50> init is complete in 4.8529925346374  1 sec


Reading root://xrootd.ba.infn.it//PATH_FILES/flatTree_TT_TuneCUETP8M2T4_13TeV-powheg-pythia8  oot
# 10000 entries, 77 branches, 8.875920295715332 MB, 0.9596493244171143 sec, 9.24912889518941  MB/sec, 10.42047313071777 kHz
# 10000 entries, 77 branches, 8.868906021118164 MB, 1.2938923835754395 sec, 6.8544386949790  MB/sec, 7.728618026459661 kHz
# 10000 entries, 77 branches, 8.869449615478516 MB, 1.1267895698547363 sec, 7.8714338974773  9 MB/sec, 8.874771534572496 kHz
--- first pass: 1003980 events, (22-flat, 52-jagged) branches, 312 attrs
<MLaaS4HEP.reader.RootDataReader object at 0x7f8410e15f90> init is complete in 4.535124778  47559 sec


write global-specs.json
load specs from global-specs.json for root://xrootd.ba.infn.it//$PATH_FILES/flatTree_ttHJetTobb_M125_13TeV_amcatnloFXFX_madspin_pythia8.root
load specs from global-specs.json for root://xrootd.ba.infn.it//$PATH_FILES/flatTree_TT_TuneCUETP8M2T4_13TeV-powheg-pythia8.root
init RootDataGenerator in 11.186564683914185 sec
```

```
label 1, file <flatTree_ttHJetTobb_M125_13TeV_amcatnloFXFX_madspin_pythia8.root>, going to read 4858 events
read chunk [0:4857] from /$PATH_FILES/flatTree_ttHJetTobb_M125_13TeV_amcatnloFXFX_madspin_pythia8.root
# 10000 entries, 77 branches, 9.52220344543457 MB, 1.3816642761230469 sec, 6.891835889507034 MB/sec, 7.237648228164387 kHz
total read 4858 evts from /$PATH_FILES/flatTree_ttHJetTobb_M125_13TeV_amcatnloFXFX_madspin_pythia8.root


label 0, file <flatTree_TT_TuneCUETP8M2T4_13TeV-powheg-pythia8.root>, going to read 5142 events
read chunk [4858:9999] from /$PATH_FILES/flatTree_TT_TuneCUETP8M2T4_13TeV-powheg-pythia8.root
# 10000 entries, 77 branches, 8.875920295715332 MB, 1.7170112133026123 sec, 5.169401473297779 MB/sec, 5.8240737873606205 kHz
total read 5142 evts from /$PATH_FILES/flatTree_TT_TuneCUETP8M2T4_13TeV-powheg-pythia8.root
```

Create the chunk

MLaaS4HEP performance

- We used a dataset with 28.5M events with 74 branches (22 flat and 52 Jagged), and a total size of about **10.1 GB**.

- We performed all the tests running MLaaS framework on **two** different platforms
  - macOS, 2.2 GHz Intel Core i7 dual-core, 8 GB of RAM
  - CentOS 7 Linux, 4 VCPU Intel Core Processor Haswell 2.4 GHz, 7.3 GB of RAM CERN Virtual Machine

- The ROOT files are read from **local** file-systems (SSD storages) and **remotely** from the Grid sites. In particular, we read files remotely from three different data-centers located at Bologna (BO), Pisa (PI), Bari (BA)

- Based on the resource we used and if the ROOT files were local or remote, we obtained:
  - **specs computing phase (chunk size = 100k events)**
    - Event throughput: 8.4k – 13.7k evts/s
    - Total time using all the 28.5M events: 35 – 57 min
  - **chunks creation in the training phase (chunk size = 100k events)**
    - Event throughput: 1.1k – 1.2k evts/s
    - Total time using all the 28.5M events: 6.5 – 7.5 hrs

- **TFaaS** benchmarks on CentOS 7 Linux, 16 cores, 30 GB of RAM
  - **500 req/sec** for TF model with 27 features, 1024x1024 hidden layers, using 5k calls with 200 concurrent clients

MLaaS4HEP cloudification with DODAS

All the material here: github.com/lgiommi/mlaas_cloud

Creation of a docker image able to run the workflow.py script

Create an Ansible playbook to automatize the configuration and deployment of the container with dependencies

Convert the Ansible playbook into an Ansible role

Creation of a Tosca template to define the resource requirements and the input parameters for the creation of the docker container

Create the deployment from command line

Run workflow.py interactively or with jupyterhub

```
dodas create lgiommi-template.yml
   dodas login <infID> <vmID>
```

MLaaS4HEP using Jupyterhub

- We provide a SaaS solution for a sharable jupyter notebook
- Token-based access to the jupyterhub, with the support for a customizable environment

DEMO

- Integrate cloud storage for managing the required files (ROOT files, ML model, etc.)

```
# . ./shared/setup_local
(base) # cd /workarea/shared/folder_test
(base) # ../../workarea/MLaaS4HEP/src/python/MLaaS4HEP/workflow.py --files=files_test.txt --labels=labels_test.txt --
model=keras_model.py --params=params_test.json
model parameters: {"nevts": -1, "shuffle": true, "chunk_size": 10000, "epochs": 5, "batch_size": 100, "identifier": ["runNo", "evtNo",
"lumi"], "branch": "events", "selected_branches": "", "exclude_branches": "", "hist": "pdfs", "redirector": "root://gridftp-storm-
t3.cr.cnaf.infn.it:1095", "verbose": 1}
Reading ttH_signal.root
# 10000 entries, 29 branches, 1.10626220703125 MB, 0.034181833267211914 sec, 32.364039645948566 MB/sec, 292.5530623775014 kHz
# 10000 entries, 29 branches, 1.10626220703125 MB, 0.022344589233398438 sec, 49.50917626973965 MB/sec, 447.53563807084936 kHz
```