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

Benchmark interactive analysis for future colliders

Adelina D'Onofrio, Elvira Rossi, Francesco Cirotto, Francesco Conventi, Orso Iorio, Antimo Cagnotta, Antonio D'Avanzo, Gianluca Sabella, Bernardino Spisso

Second ECFA Workshop on e+e- Higgs/EW/Top Factories, October 11-13 2023, in Paestum (Salerno)

Outline

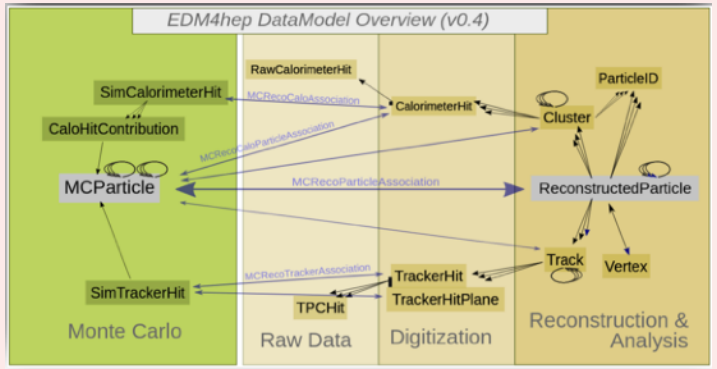
- Motivations
- The INFN Naples infrastructure
- Use case example, in the FCCee context
- Scalability results
- Towards an HTCondor model
- Conclusions

Motivations

- Most of the LHC searches/measurements rely on locally developed scripts that process the datasets, with parallel tasks and on an asynchronous batch system
- **Challenges of the future** e^+e^- colliders are pushing to **re-think the HEP computing models**
  - Impact on several aspects, from software to the computing infrastructure
- From the software perspective, **interactive/quasi interactive analysis** is a promising paradigm
  - User-friendly environment
  - The implementation is simplified by adopting open-source industry standards: *Dask, Jupyter Notebooks* and *HTCondor*
  - Validating new frameworks (e.g. *ROOT RDataFrame* with multi-threading)
- **Preliminary feasibility studies** have been pursued on **FCCee pseudo-data,** exploiting **INFN Napoli** analysis Facilities (**AFs**)
  - Distributed infrastructure which leverages *Dask*

Workflow

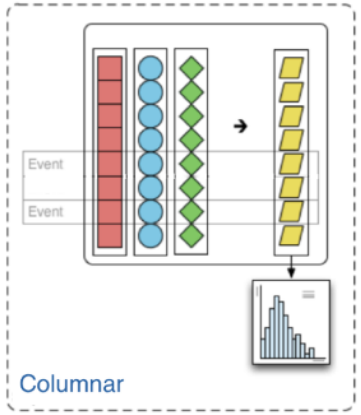
EDM4hep input data format



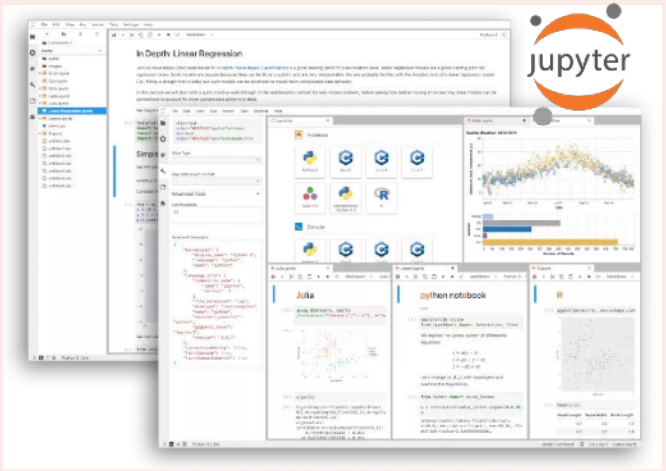
flat input ntuples

RDataFrame

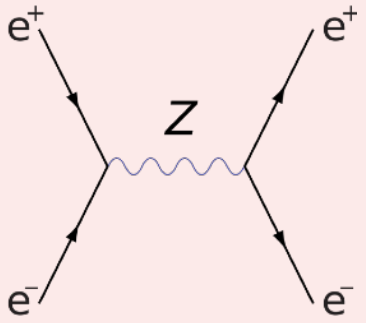
+ dask used as backend



New approach to data analysis



Use case



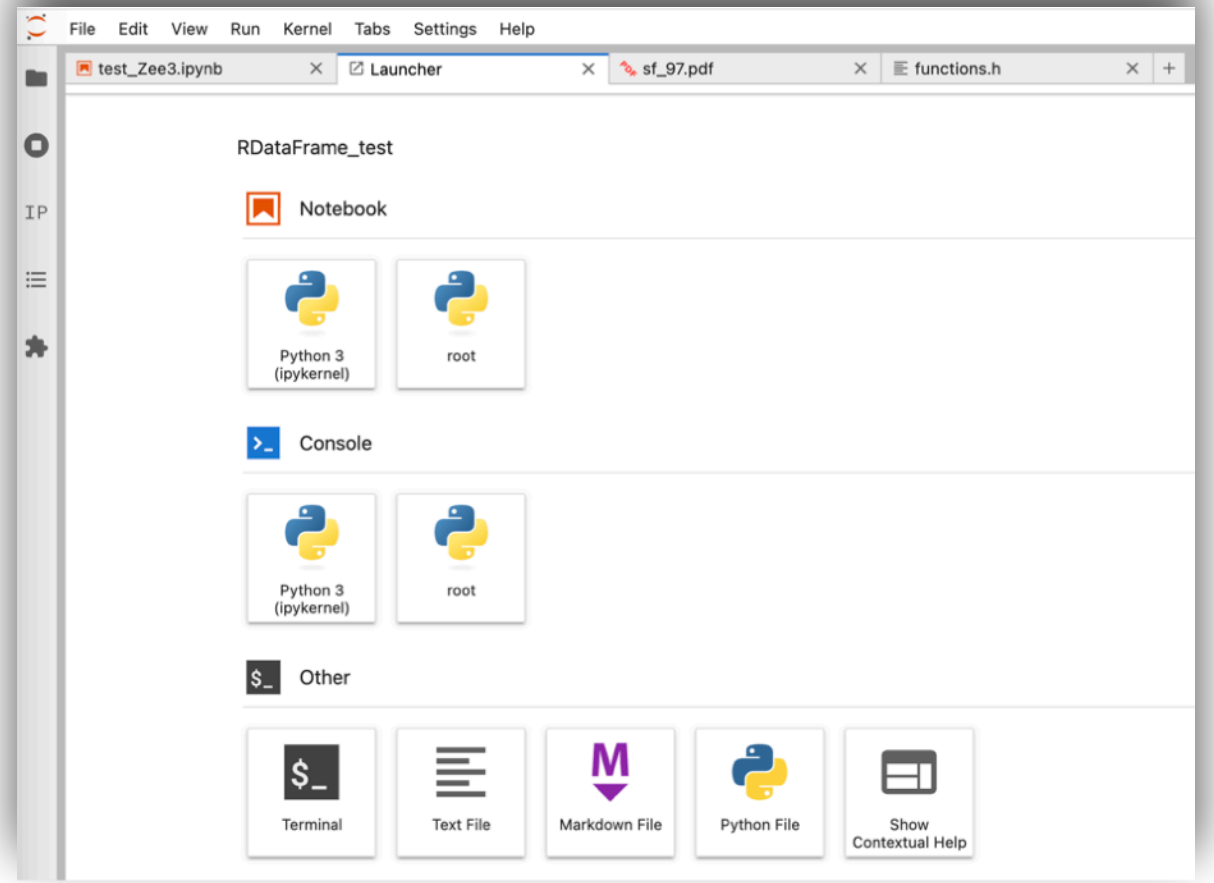
Feasibility study & Preliminary performance evaluation

INFN Napoli infrastructure

- Our group developed a local testbed infrastructure in INFN Naples (Italy)
- The local deployment is based on the *Open-Stack IaaS* paradigm
- Starting from the already existing *I.Bi.S.CO* installation, several updates were performed
- The cluster is made up of 2 identical virtual machines, each equipped with 1CPU quadCore and 8GB RAM, currently expanded up to 12 cores and 64GB
- Rocky Linux 8.6 is the operating system
- 2 nodes are equipped with **Docker** (20.10) for containerisation and **Kubernetes** (1.26.3) for the orchestration
  - 📌 One node plays as controlplane. etcf & worker; the other node acts as a plain worker
- The cluster is equipped with **JupyterHub** & **JupyterLAB** where the user can play with **Python, ROOT & Dask** libraries

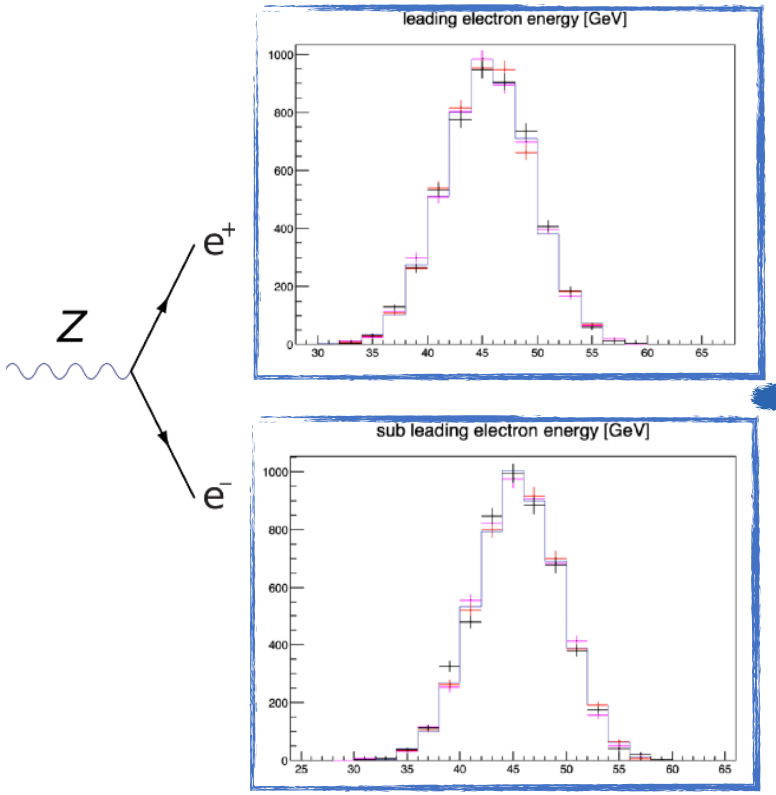
Efficient & user friendly infrastructure

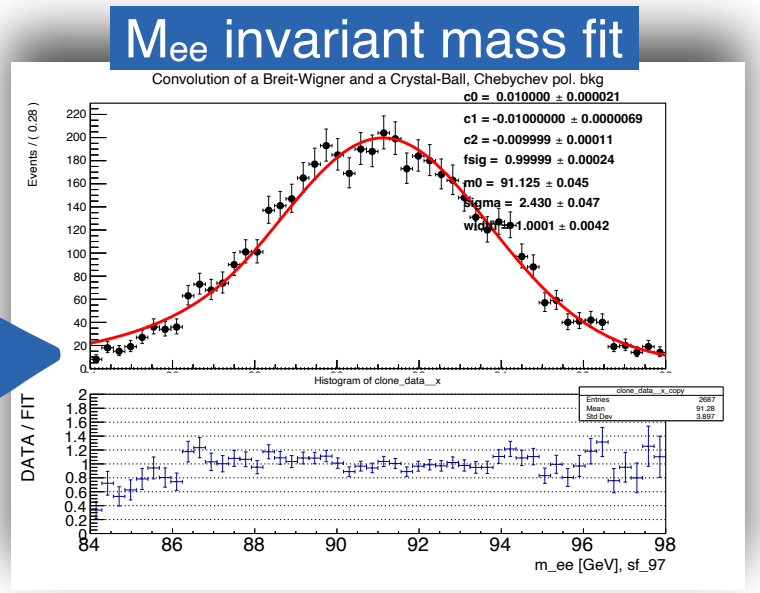
- Python & ROOT (v 6.28) kernels available
- Terminal
- Notebook implementation
  - Completely exportable and replicable



Use case

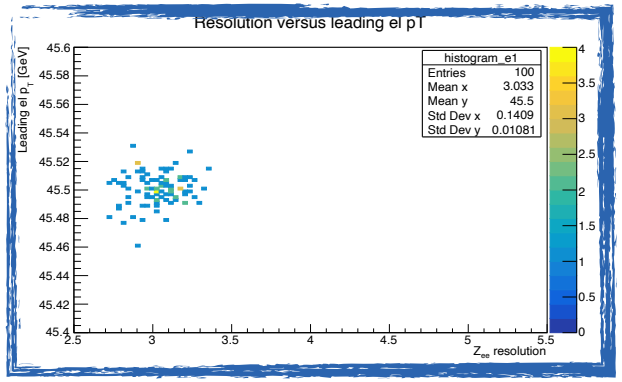
- FCCee simulation: /eos/experiment/fcc/ee/tmp/ee_Z_ee_EDM4Hep.root
  - 5k events, scaled to 1M events replicating the available dataset
  - Mimic systematic variations, gaussian smearing the electrons energy to compute M_{ee} resolution

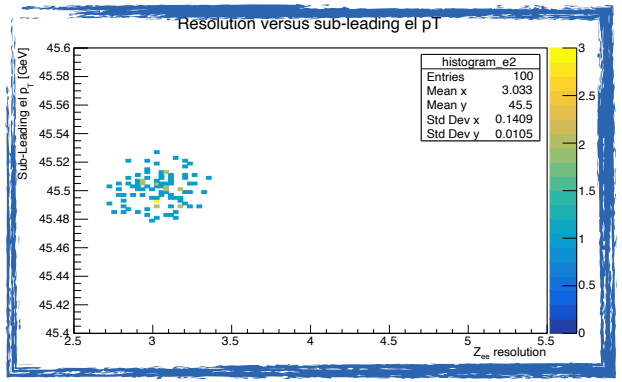




Selection and histogramming interactively via RDataFrame on JupyterHub

github link to the code





Local *vs* distributed approach

Scaling without changing your code

```
from dask.distributed import LocalCluster, Client
if distributed == True:
    RDataFrame = ROOT.RDF.Experimental.Distributed.Dask.RDataFrame
    ROOT.RDF.Experimental.Distributed.initialize(my_initialization_function)
else:
    RDataFrame = ROOT.RDataFrame
    my_initialization_function()
```

Distributed

Local

No changes required to the rest of the code

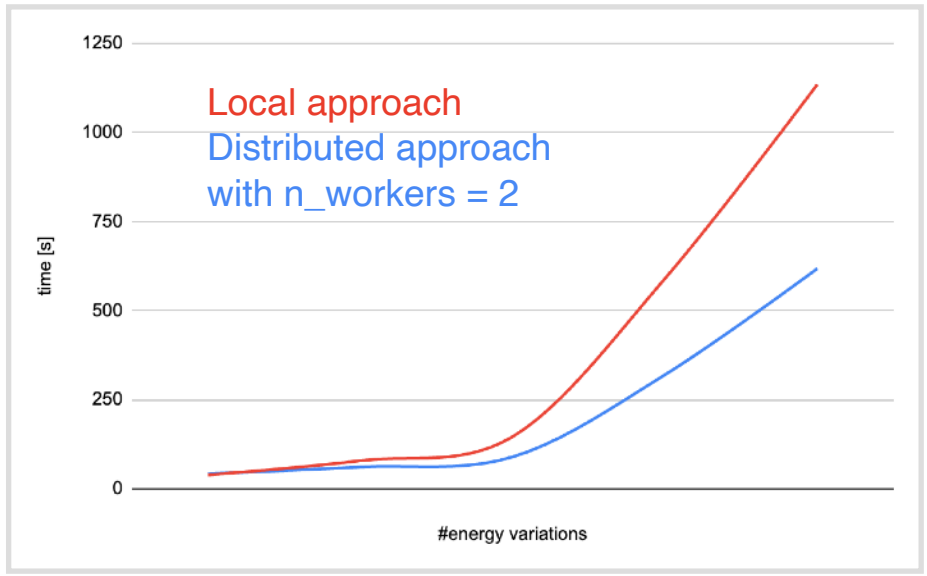
```
df = df.Define('w_nominal', '1')
df = df.Define("m_e","0.0005124") #GeV
df_ge = df.Define("goodelectrons", "Particle.charge[0]*Particle.charge[1] < 0.").Filter("goodelectrons > 1")
```

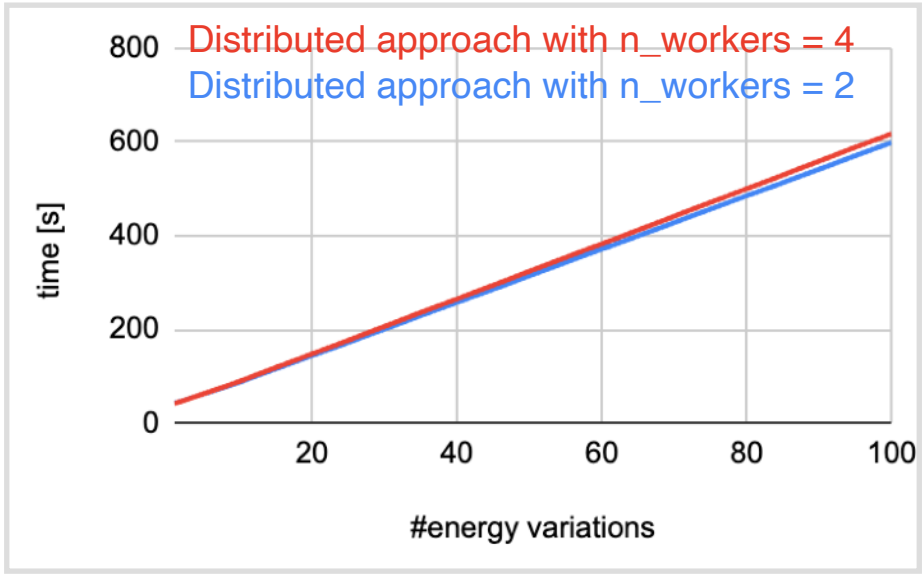
How to compare the performance?

Defined Metric	
Overall execution time	Time elapsed from the start of the execution (execution triggered) to the end of execution

Local *vs* distributed approach

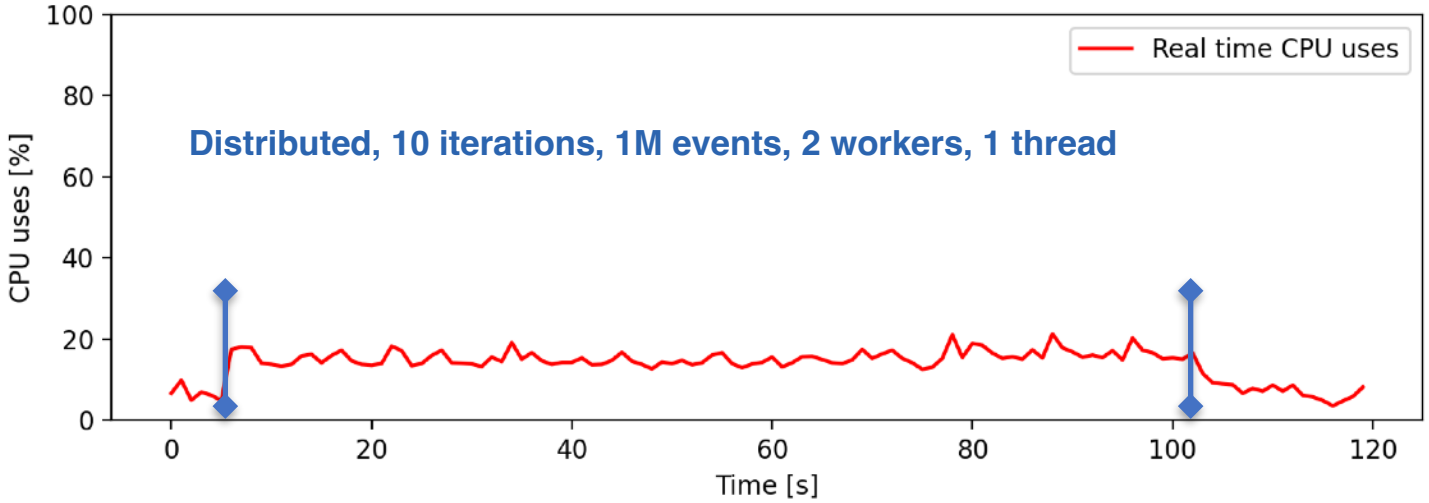
```
cluster = LocalCluster(n_workers=2, threads_per_worker=1, processes=True)
```



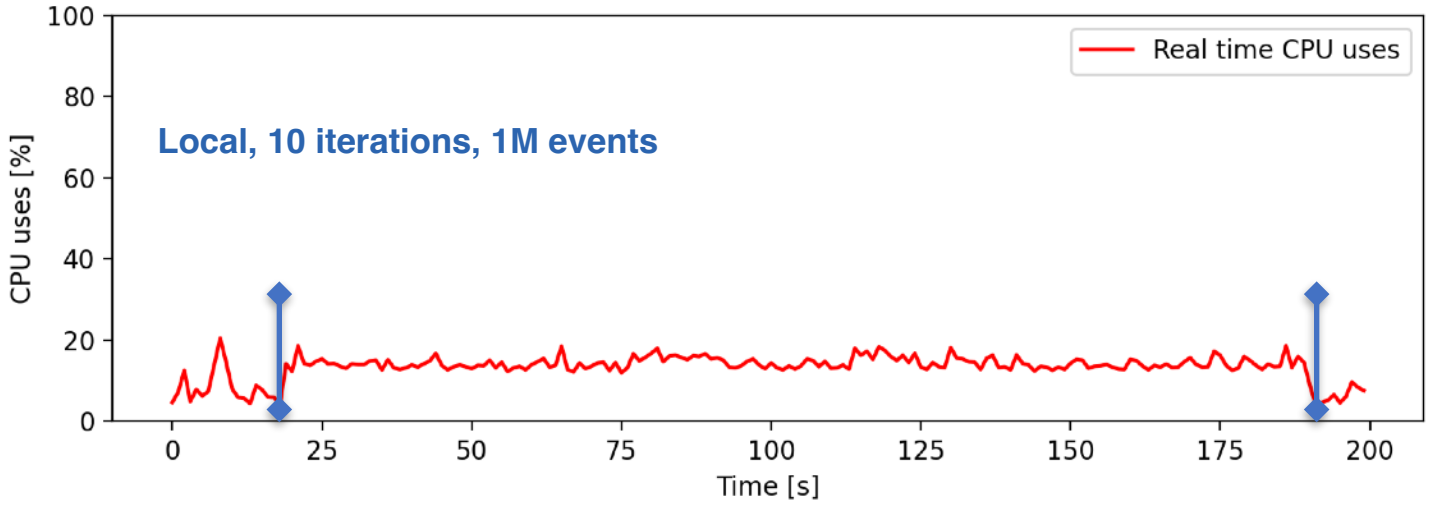


- Exploiting the distributed approach, the execution time halves wrt the local approach if we iterate over a significative number of energy variations (> 10)
- Changing the number of workers from 2 to 4, the execution time is stable

CPU usage?

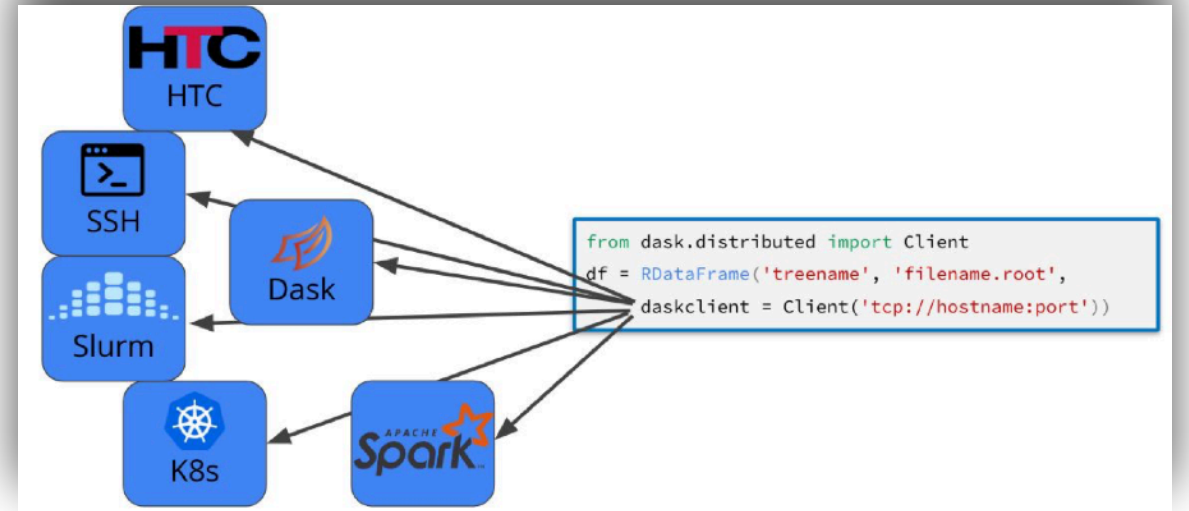


- Comparable CPU usage
  - Due to the simple use case tested



Towards a Dask + HTCondor model

- Based on INFN Perugia *analysis facility*
- Introducing HTCondor queues, the performance improves by a factor 2
- Increasing the number of workers is beneficial when running on many iterations



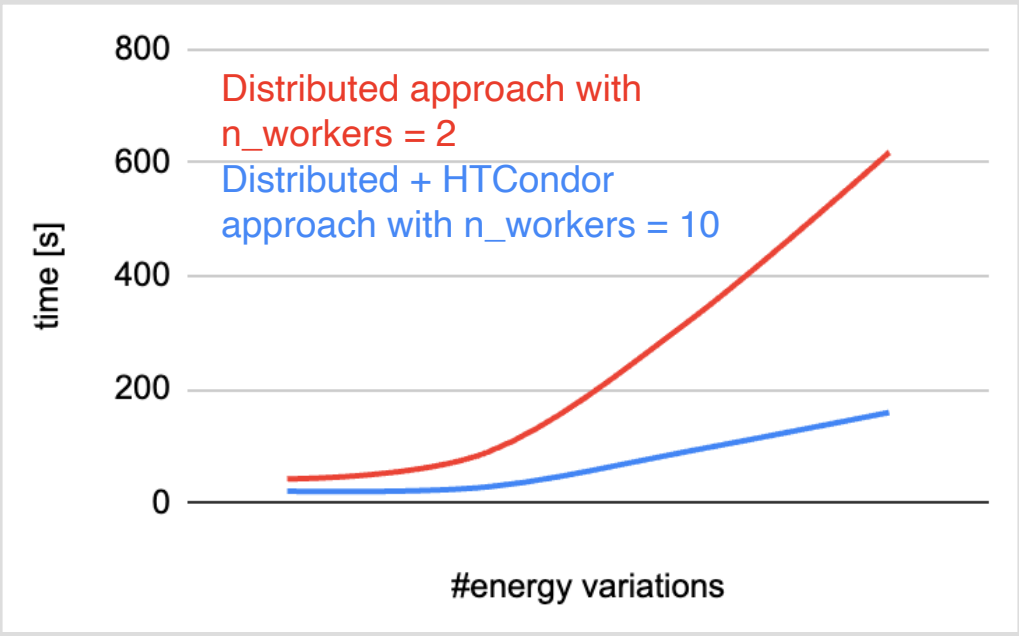
n_workers = 2

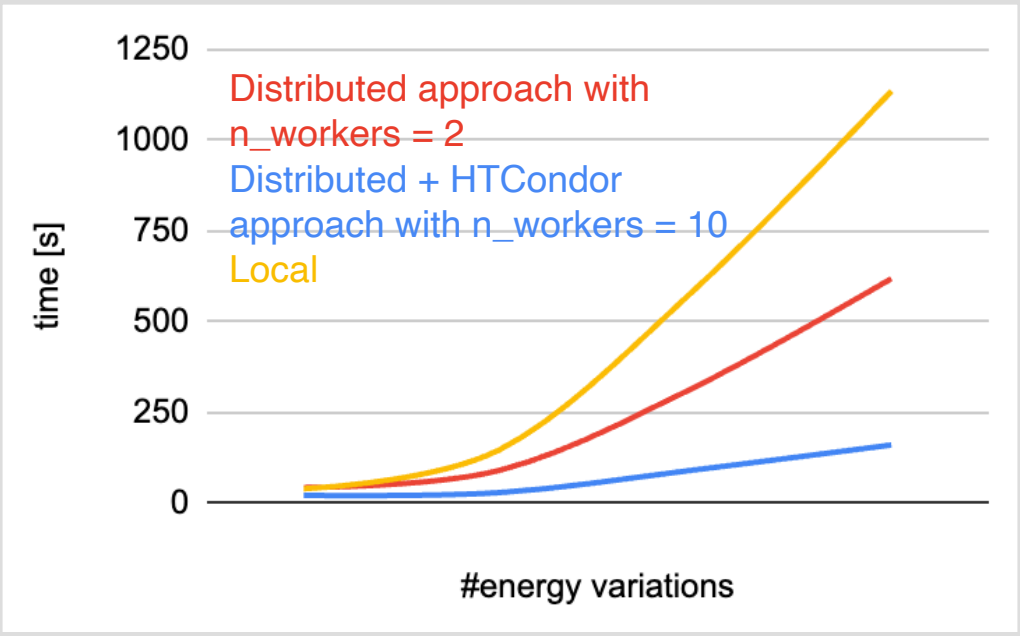
# iterations	Dask+HTCondor	Dask
1	22.96 s	42.02 s
50	258.35 s	320 s
100	497.71 s	618 s

Dask + HTCondor

# iterations	n_workers = 2	n_workers = 10
1	22.96 s	20.36 s
50	258.35 s	90.89 s
100	497.71 s	159.26 s

Towards a Dask + HTCondor model





- Exploiting the distributed approach, the execution time halves wrt the local approach
- Moving to a Dask+HTCondor model, we gain up to another factor 2
  - Increasing the number of workers, the execution time further improves

Conclusions & Next Steps

- Interactive analysis feasibility studies on the INFN Naples infrastructure succeeded
- Towards an INFN national cloud infrastructure with a datalake model to facilitate future analyses (hopefully starting from LHC Run 3)

➡ **Short term goals:**

📌 Dask interface with HTCondor queues on the INFN Naples facility

➡ **Medium-long term goals:**

📌 A single HUB for the data analysis: web based & framework agnostic

📌 Kubernetes + Dockers: allow the usage of images both locally and over all the distributed resources

Thank you!