# HOW19 Trip Report: Data Analysis and PyHEP

Danilo Piparo (EP-SFT)

# This Talk

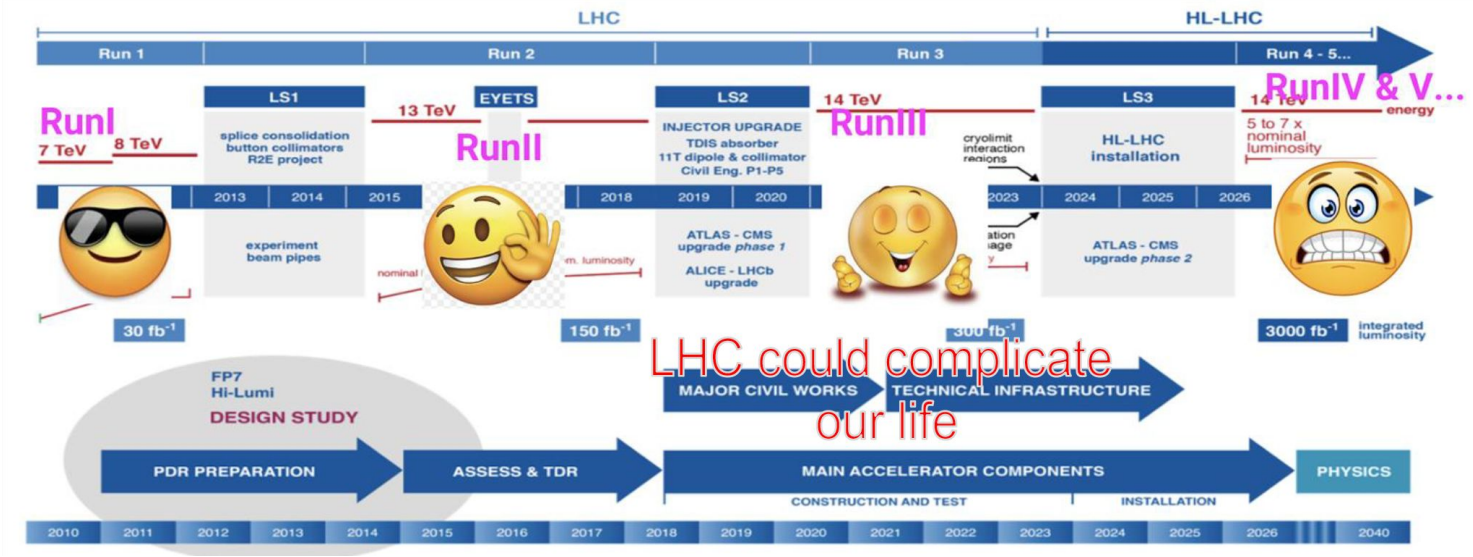
- Covers the Data Analysis Working Group and PyHEP sessions at HOW2019
- My take on the most relevant information and trends

# The Sessions

- Data Analysis Working Group (DAWG) sessions: 2 x 1.5h, 6 talks in total
  - Plenty of discussion time allocated
- PyHEP: 1.5h, 5 talks (one by J. Helmus, Anaconda Inc.)
- Diverse audience: universities, labs, professors, students, scientists of all ages, LHC and outside, analysis physicists, technology experts.
- In general, a good showcase of technologies, packages and ideas around.

# Exciting Times!

- Feeling somehow similar to the years preceding LHC start
- Opportunities, opportunities, opportunities!
  - Also in the field of data analysis



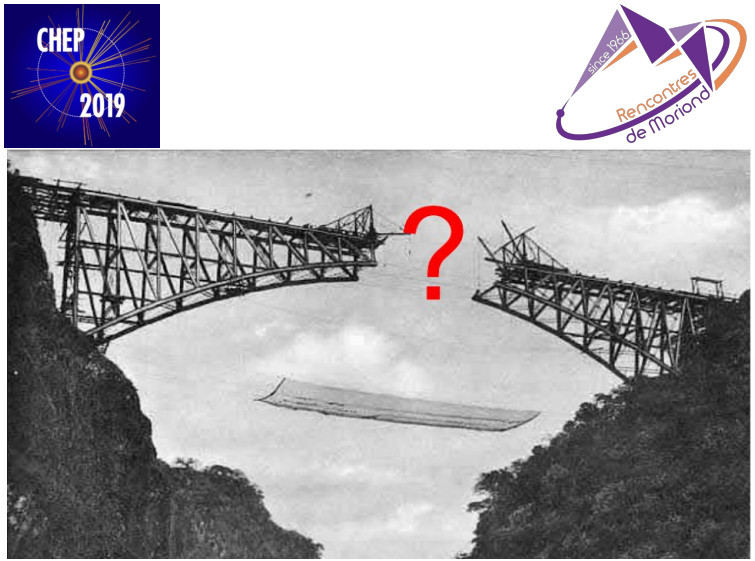

T. Boccali, M. Klute

# Data Analysis Working Group (DAWG)

P. Laycock

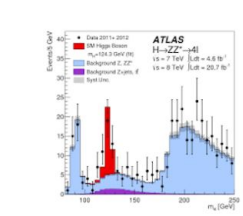
# DAWG Objective

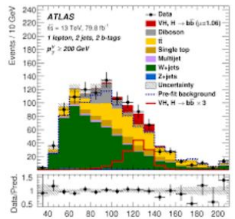
- Building bridges between communities
  - E.g. People attending Moriond and people attending CHEP

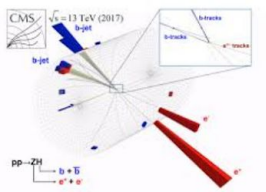



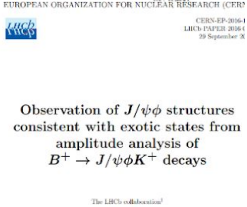







Inspired by A.Rizzi

# Some Identified Issues of Analysis - I

- Analysis is always performed in a rush:
  - No validation
  - Wild cut and paste and reuse code meant for other goals
  - Prolonged code life (developers gone, try to keep it running)
- Multiple frameworks all doing mostly the same things:
  - Cost of maintenance multiplied by N frameworks
  - Number of bugs flying around multiplied by N frameworks
  - Half baked solutions: nice features of fwk A not in fwk B and viceversa
- Analysis preservation
  - Can we complement the full dump of some analysis working folder in a docker container?

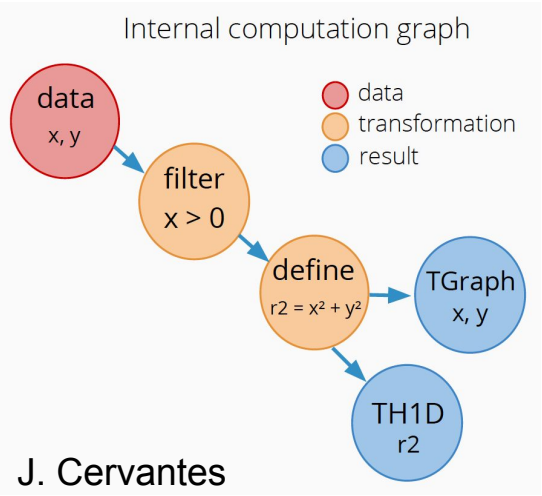
# Some Identified Issues of Analysis - 2

- Approaches which may become unaffordable in the future:
  - Compute everything I would possibly need
  - Store everything I would possibly need
  - Leave it there forever, even if I stop using it
  - Never ask myself how much does it cost


- We'll need efficient backends but **physicists cannot and will not always write optimised analysis code**
- Can we improve providing high quality trainings?
  - How can we make analysis simpler?

# Some Proposed Solutions

- Declarative analysis is seen as part of the solution
  - Specify the what, not how to achieve it
  - Implementations available already: Coffea, RDataFrame, LINTTOROOT, BASF
- Converge towards common interfaces for non-event data
  - ROOT mentioned explicitly as a potential vector for such tools
- Strive for "Framework Efforts" across collaborations
- Do not push to users half-baked solutions
  - Identify set of high level, realistic (e.g. systematics) benchmarks of increasing complexity




J. Cervantes

N. Smith

```
ele = electrons[(electrons.p4.pt > 20) &
                                    (np.abs(electrons.p4.eta) < 2.5) &
                                    (electrons.cutBased >= 4)]

mu = muons[(muons.p4.pt > 20) &
                                (np.abs(muons.p4.eta) < 2.4) &
                                (muons.tightId > 0)]
```

# What Triggered Discussion?

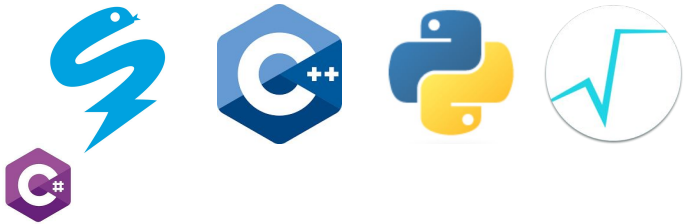
The kind of observations & questions to the audience which triggered discussion

- Which kind of flexibility we value most? (trains vs reduced common formats)
- Convergence on Jupyter notebooks as analysis platform, hiding the how is good

**Many constructive and quite open x-experiment chats**

# Languages

- The C++-Python duo is the reference
  - Functionality-, performance- and programming model-wise
- Clear trend: propose Python to physicists and accelerate it with C++/Python jitting and bindings to compiled libraries
- An example of C# (+LINQ)
  - Can we re-propose the useful concepts discussed w/o imposing the language itself?
- No in-depth discussion about this but the idea of an Analysis Description Language is in the air.



## Optimization of Signal Selection

**Analysis of the *n*-tuples is done with Python:**

- *Pandas* and *numpy*
- *root_pandas* or *uproot* to load ROOT files
- *scikit-learn* or basf2 MVA package for MVA methods
- *matplotlib* for plots
- convert *n*-tuples to hdf5 files (these are loaded $\sim$10 times faster)
- data analysis in *jupyter notebooks*

**Why Python?**

- Well documented!
- Easy to integrate into the rest of the analysis
- Modern and nice interface...

F. Metzner (Belle II)

# Looking For Alternatives - I

## *2011 "How much analysis can I do and not touch ROOT?"*

- Frustration with how much was hidden, unexpected behaviors (ROOT/RooFit)
- Changes in versions/Installation issues
- Wanted to engage non-particle physics students with more general-use tools
  - Outreach efforts
- Factorized ROOT functionality (e.g. file IO separate from plotting) (***pre-uproot***)
- *Green field* to play in

M. Bellis

- File read is....different
  - h5hep reads everything into memory which is super fast!
  - Looping over events is different, but script performances were still faster with h5hep, sometimes significantly (up to 2x faster)
  - Can load subsets of the datasets (variables) or subsets of the events

# Looking For Alternatives - 2

- Integrate seamlessly with data, plotting and statistic packages

zfit
scalable pythonic fitting

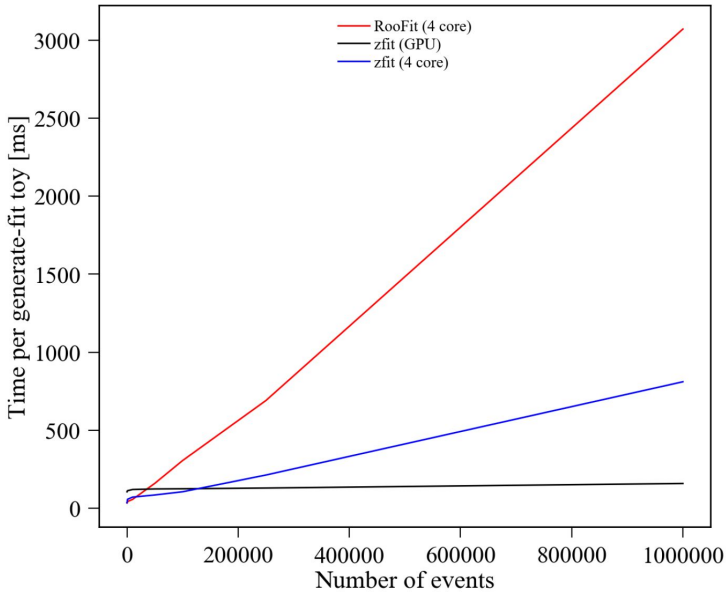
- Uses uproot for ROOT files (no ROOT dependency!)
- Used by lauztat for statistics
- Can be used by phasespace for resonance modeling

```
obs = zfit.Space("x", limits=(-10, 10))

mu =    zfit.Parameter("mu",          1,  -4,  6)
sigma = zfit.Parameter("sigma",       1, 0.1, 10)
lambd = zfit.Parameter("lambda",     -1,  -5,  0)
frac =  zfit.Parameter("fraction", 0.5,   0,  1)

gauss = zfit.pdf.Gauss(mu=mu, sigma=sigma, obs=obs)
exponential = zfit.pdf.Exponential(lambd, obs=obs)

sum_pdf = zfit.pdf.SumPDF([gauss, exponential], fracs=frac)

sum_pdf = frac * gauss + exponential
```

parameters

models

equivalent



# Conda & ROOT

- Open source package and environment management system
- ROOT packaged for Conda: 3 lines to create a Conda environment and get ROOT

```
conda create -n myrootenv python=3.7 root -c conda-forge
conda activate myrootenv
conda config --env --add channels conda-forge
```

C. Burr, E. Guiraud,
H. Schreiner

**Nice collaboration example**

tcanvasmagic: Demo of new proposed notebook magic for ROOT.

launch binder

environment.yml

```
channels:
  - conda-forge
dependencies:
  - root
```

# Conclusions

- HOW: diverse profiles, enriching discussions among experts and experiments
  - Showcase of current activities and solutions available
- Exciting times ahead of us!
  - Opportunity to become even more active & curious & courageous
- The DAWG started to identify directions to improve HEP data analysis and echoed them back to HSF
  - Solutions to some problems being prototyped *now*!
- Showcase of available solutions may evolve in competition
  - Talk, communicate, be open
  - Identify benchmarks, metrics and cost models (Open Data as starting point?)
  - Avoid half-baked solutions