

# Tuesday.

Analysis Ecosystem WS  
Amsterdam, 2017-05-22..24

# Vision

- Hardware: Fons
- KPMG: Max Back
- Visions: Eduardo Rodrigues, Gordon Watts, Gerhard Raven
- ROOT: TDataFrame, general

# Fons: Openlab / Hardware

- Past mis-predictions (Itanium, Objectivity) - HEP needs to own its fundamental ingredients
- Intel 3D XPoint “persistent RAM”: 1000x lower latency than SSD at price of DRAM
  - no cache, random access just fine (data layout!), bottleneck moves to CPU (ouch, Java)
- Plentycore, example of Intel brain growth challenge
- Hybrid CPU: FPGAs, compression ASICs

# Fons

- Warns against dumping all the assets we have:
  - efficiency, performance - why interpreted and slow data format when we **are** fast?
- Must advertise ROOT outside HEP
- Long-time programs require us to focus on lang-time support. And we **have** that (in contrast to other sciences)
- Performance-centric projects turn to C++ and we are already there; “grass not always greener on the other side”

# Max Baak

- Open Source tools allow everything data analysis
- “HEP has obligation to use + contribute to existing open-source tooling”
- ROOT: slimmer, easy install, focus on key + unique parts (e.g. uncertainty, RooFit); don't reimplement existing, world-dominating tools; make python a 1st class citizen, embrace python-based layers

# Eduardo Rodrigues

- Evolving too slowly (ML)
- SWAN limited to tutorials etc, won't scale to analysis
- Focus on interoperability between tools; Grid tools still not good enough / scaling
- LHCb Turbo Stream analysis is the way to go
- SciKit-HEP, Everware

# Gordon Watts

- Declarative approach:  
`events .Select(e =>  
 e.Data.eventWeight) .FuturePlot("event_weights",  
 "Sample EventWeights", 100, 0.0,  
 1000.0) .Save(hdir);`
- C#-based LINQ separates backend engine. Allows
  - detection of which parts changed / caching
  - storing query producing a plot / reproducibility
  - reordering of filters to optimize

# Gordon (2)

- All-in-one(-file) analysis, no A drives B with tool C
- Strongly typed (with types derived from data)
  - enables compile-time errors, tooling (IDE / completion)
- Driven by Jenkins, checking analysis into repo, caching brings this from 7hours to 2mins
- TDataFrame might be on the right path (and Gordon provided early feedback!)



# Gerhard Raven

- Agrees with Gordon:
  - declarative analysis
  - use simple layer to drive complex backend (but increases the gap between physicists and computing knowledge)
- Separate “what” (isolated electrons etc) from how: simpler analysis specification, better performance due to optimizing backend

# Gerhard (2)

- Data provenance is an issue (“pT”?)
- Funding agencies start to demand reproducibility!
- Collaborate with outside tools; if we rely on them, own them!
- Can only convince of new analysis scheme by showing full chain

# Pere: TDataFrame

- Declarative, as requested by Gordon and Gerhard
- Knows about ROOT I/O, optimized backend
  - could re-implement backend, using llvm optimizers e.g. to do more CPU work to reduce data reads
- Python bindings (with discussions on allowing python-lambdas)

# Axel: Future ROOT

- Still “ROOT”: works well, HEP owns its key tools
- Focus on key ingredients: I/O, Math/Hist, Graph, Foundation and glue (PyROOT on top of cling!)
- Make it efficient (wrt to physicists, not CPU): no debugging; concurrency under the hood; let ROOT provide efficient glue to e.g. numpy
- Make it simple: install; interoperability (read HDF5); simple formulation of analysis with obvious interfaces

# Personal Workshop Summary

# The Goal vs Reality

- Get an idea of the future of analysis
  - ... to propose projects to funding agencies
- Main themes: python, machine learning, getting rid of parts of ROOT
- I.e. find a cozy corner to shine, develop, then (by virtue of “HSF context”?) expect generic adoption
- Very little (0?) “let’s help ROOT” help us

# Axel's Prediction

- Papers and write-ups
- “Community-embraced” “projects” with little communication with ROOT
- Repositories with names of permutations of “hep”, “py”, “analysis”, i.e. claiming world domination
  - good: funding! bad: relevance?
- ROOT needs to do what ROOT needs to do

# Axel's Conclusion

- Ensure PyROOT support, put it on top of cling
- Offer good tools that are used by the masses
  - focus on technical solutions, e.g. TDataFrame
  - don't reimplement but provide efficient bridges; communicate why our pieces are fundamentally better, or deprecate them
- Apparent disconnect - regular reviews!