ROOT's Future

Axel Naumann
2018-09-18, JLAB Computing Round Table

Outline

- Context: ROOT's role in the past, now and in the future
- Consequence and vision: what to expect from ROOT (and what not)
- Progress: where we are and what we do
- Conclusion

Examination of Analysis Software in HENP

- ROOT is a centerpiece of HENP
  - virtually every physicist uses ROOT for analysis
  - > 1 exabyte of data in ROOT format
- Common (also graphics) language, common data format, common grounds
  - allows physicists to switch between experiments; enables tools applicable across experiments
  - Coherently designed, integrated solution with optimized interplay

Examination of Analysis Software in HENP (2): non-ROOT

- Alternative open source "big data" software is now production grade
  - but in many aspects, "big data processing" != physics analysis
    - coding analysis; usability; CPU efficiency; data delivery; setup-cost / scalability; event-based; must not skip data points;...
- Options are adapt us to them or them to us
  - but see Google, Facebook influencing development by *driving* it
  - "adapt them" considerable dev + maintenance, one end not under control

Examination of Analysis Software in HENP (3): in-House

- Common tools = common knowledge: expertise within the community, e.g. math, data format, visualization
- Allows to predict changes, adapt and benefit: targeted solutions (with tailored R&D) to our very own problems
- Single point of improvement: contribute here to have an impact, coherency and synergies (experiment vs analysis etc) guaranteed
- Interface with and learn from other tools: become part of the industry landscape, rather than suffer from efficiency hit due to non-HENP tools that are "almost good fits" or work around it

Summary: ROOT's Role in Past, Present and Future

- Past: ROOT became *the* tool for HENP data analysis
- Present: same, but computing-oriented physicists explore non-ROOT territory
- Steer ROOT to remain competitive with alternative solutions
  - invest in development, public benchmarking, smart R&D
  - use existing tools for non-owned periphery (e.g. libclang, OpenUI)
- Advantage: community knows its challenges; gets a coherent, reliable, performant and agreed solution

Consequence and Vision

Team

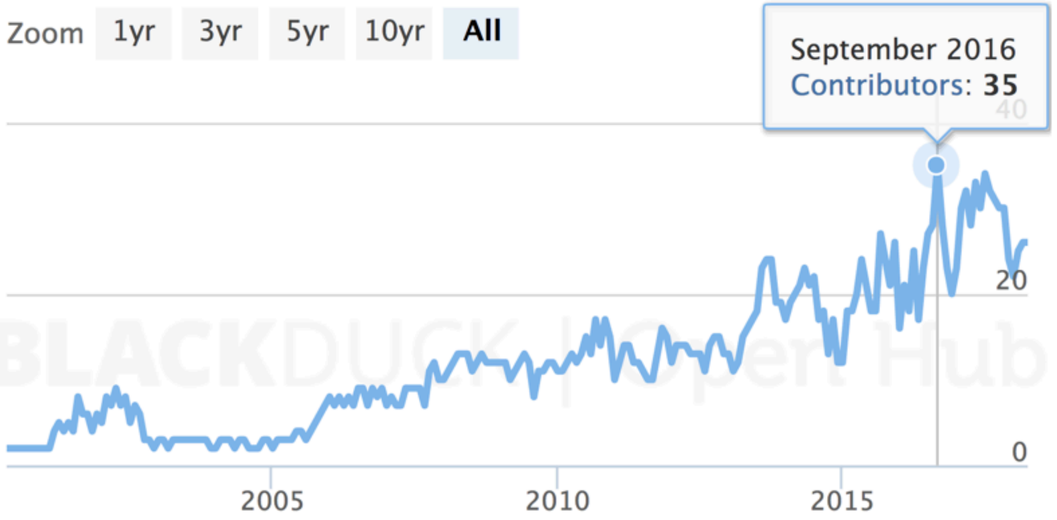
- Kim Albertsson, CERN
- Guilherme Amadio, CERN
- Sitong An, CERN
- Bertrand Bellenot, CERN
- Iliana Betsou, CERN
- Philippe Canal, Fermilab
- Olivier Couet, CERN
- Enrico Guiraud, CERN
- Sergey Linev, GSI
- Lorenzo Moneta, CERN
- Alja Mrak Tadel, UCSD
- Axel Naumann, CERN
- Danilo Piparo, CERN
- Oksana Shadura, Uni Nebraska Lincoln
- Matevz Tadel, UCSD
- Yuka Takahashi, CERN+Uni Cincinnati
- Enric Tejedor, CERN
- Xavier Valls, CERN
- Vassil Vassilev, Princeton Uni
- Stefan Wunsch, CERN

Contributions

- Many, many part-time contributors
- Extremely active also due to them!
- Very sustainable dev model, for decades

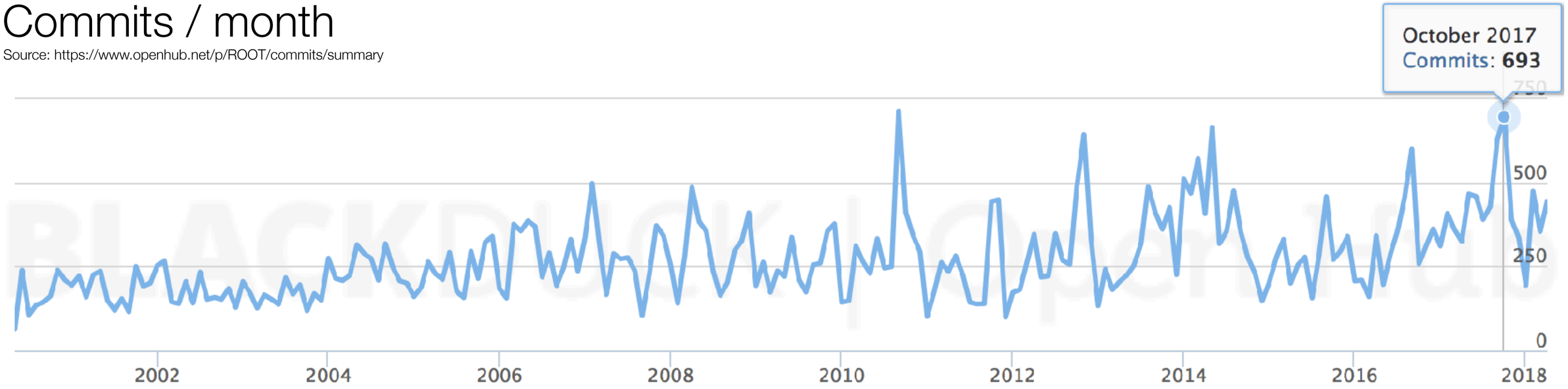
Contributors / month

Source: https://www.openhub.net/p/ROOT/contributors/summary



Commits / month

Source: https://www.openhub.net/p/ROOT/commits/summary



Investing in Development?

- ROOT dev's daily life:
  - support
  - develop small features as time allows
- Then came ROOT6 with cling
  - proved to be able to invest in multi-year dev targets, and deliver
  - within the current project, but at a cost (ROOT6: no more CINT bug fixes)

Focus

- We want to support and maintain
- We cannot implement every useful feature *and* do major development
- Focus on big developments that matter *for users*
  - no next rewrite of the build system, no 101st graphics option

“People think focus means saying yes to the thing you've got to focus on. But that's not what it means at all. It means saying no to the hundred other good ideas that there are. You have to pick carefully.”

–Steve Jobs

Conclusion: what to expect from ROOT and what not

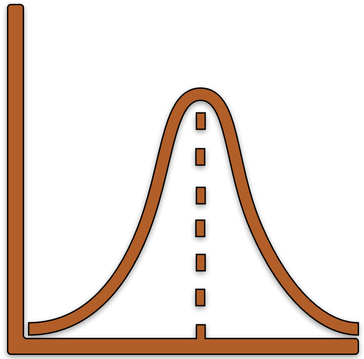
- Support remains the key ingredient of our work
- Focus on key areas instead of "small feature" growth; even contributions might not get integrated
  - histograms
  - web-based graphics / GUI / event display
  - RForest, new TTree-like container
  - RDataFrame, TMVA, Python, Parallelize

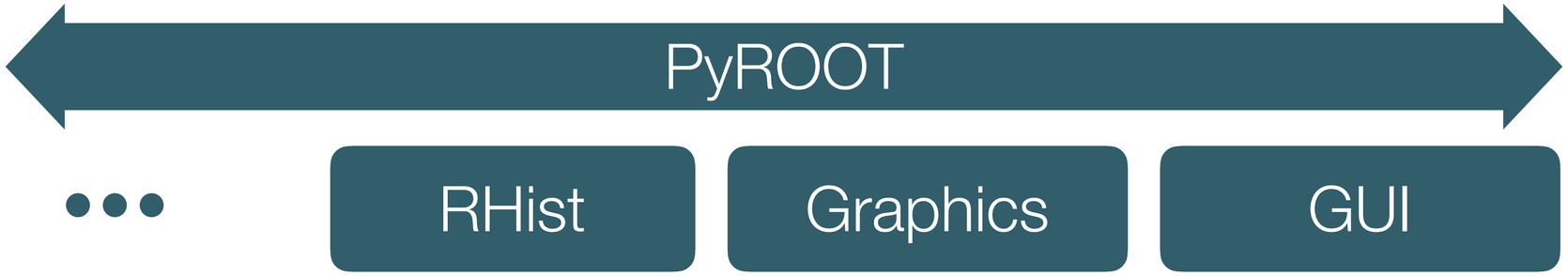
"ROOT7"
(because everything worth talking about needs a name)

Why, why those?

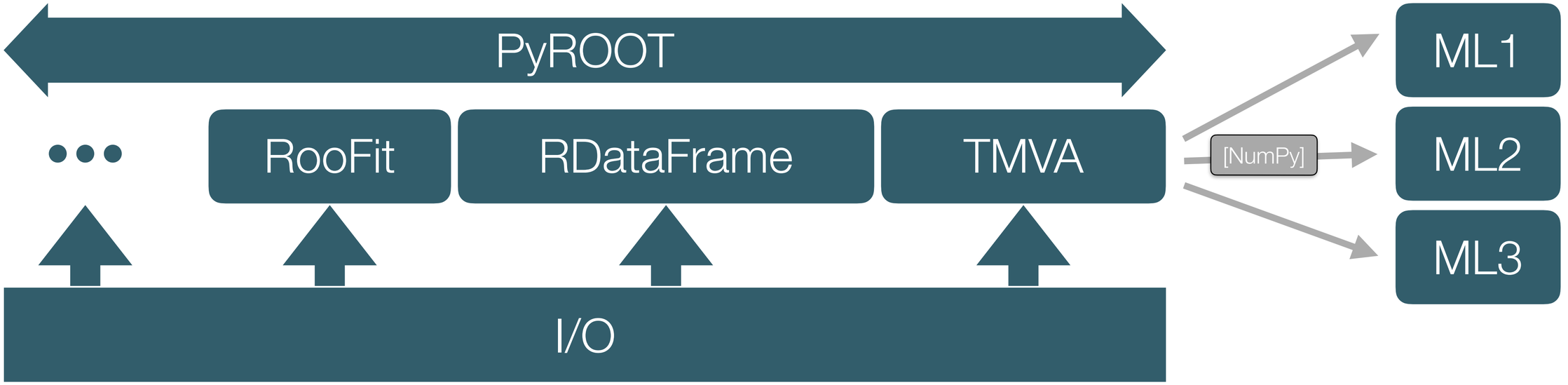
- Physicists invest in working around ROOT's past personality
  - costs time + effort, both writing analysis and education
- Community and computing context evolved, time to take stock and change some defaults
- Histograms, WebGUI/Eve, RForest, RDataFrame/TMVA/PyROOT/ Parallelization are the main parts of ROOT, visible to every physics analysis
- Ensure homogenous, consistent design, create cross-benefits

Architecture: result / interaction dimension





Architecture: data dimension



Progress: Where we are and what we do

ROOT Users' Workshop

- 110 participants
- Good overview of ROOT's current state
- Discussing analysis physicists' / experiments' feedback
- Industry presentations: Chandler Carruth / Google (large scale C++), P. Muessig / SAP (OpenUI5), S. Corlay / QuantStack (xtensor)

Histograms: ROOT::Experimental::RHist

- Speed up by moving conditional branches to compile-time
- Massively reduced interfaces by separation of concerns: storage, binning, buffering, graphics attributes, internal operations
- Remove anachronisms: named objects, pointers, TObject, virtual functions
- Extensible: uncertainties, algorithms
- Looking for early adoption to guide development!

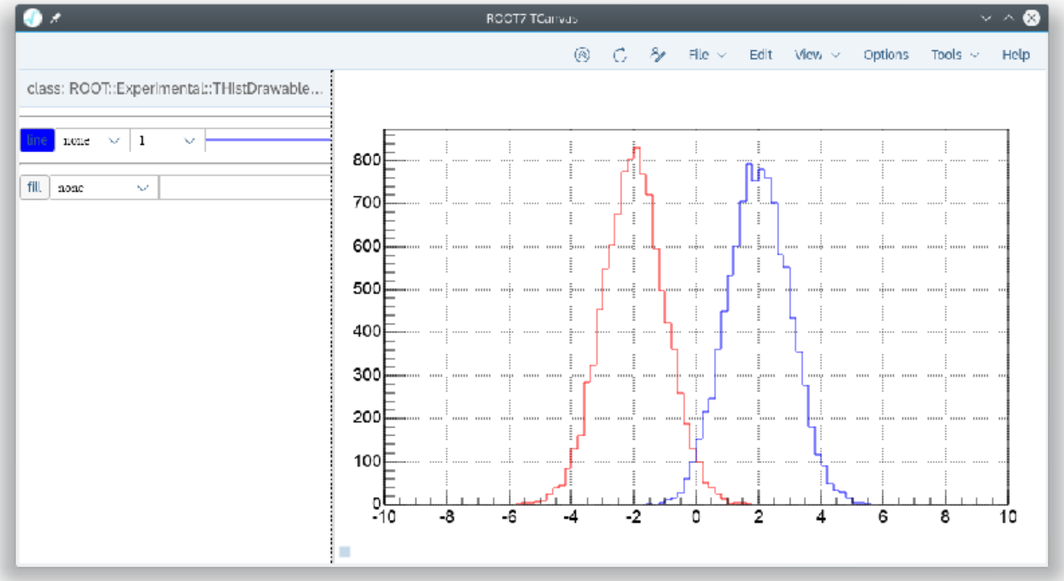
```
// Create a 2D histogram.
// X axis equidistant bins, y axis irregular binning.
Experimental::RH2D hist({100, 0., 1.},
                        {{0., 1., 2., 3., 10.}});

// Fill weight 2.1 at the coordinate 0.01, 1.02.
hist.Fill({0.01, 1.02}, 2.1);
```

WebGUI, WebGraphics, WebEve

- HTML5 + CSS + JavaScript, using existing libraries: three.js, D3.js, OpenUI5
- Prototype development in ROOT::Experimental
  - graphics painters, in the context of JSROOT, painting of v6 and v7 objects, for instance histograms and primitives: root --web
  - GUI: fit panel, work on RBrowser
  - Eve: a first geometry + track viewer + editor

Sergey: JSROOT
Sergey: WebGUI

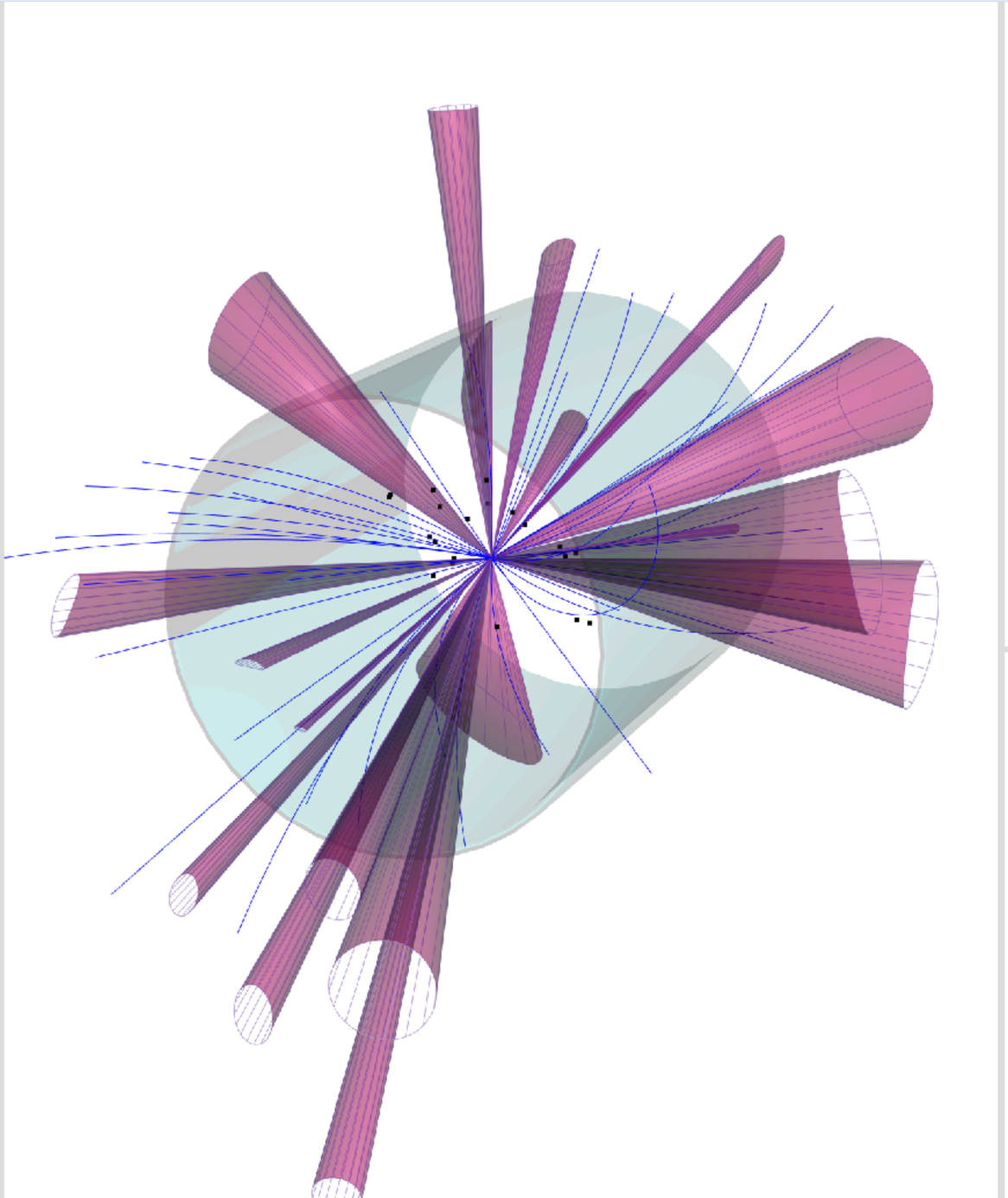
Alja: Eve7

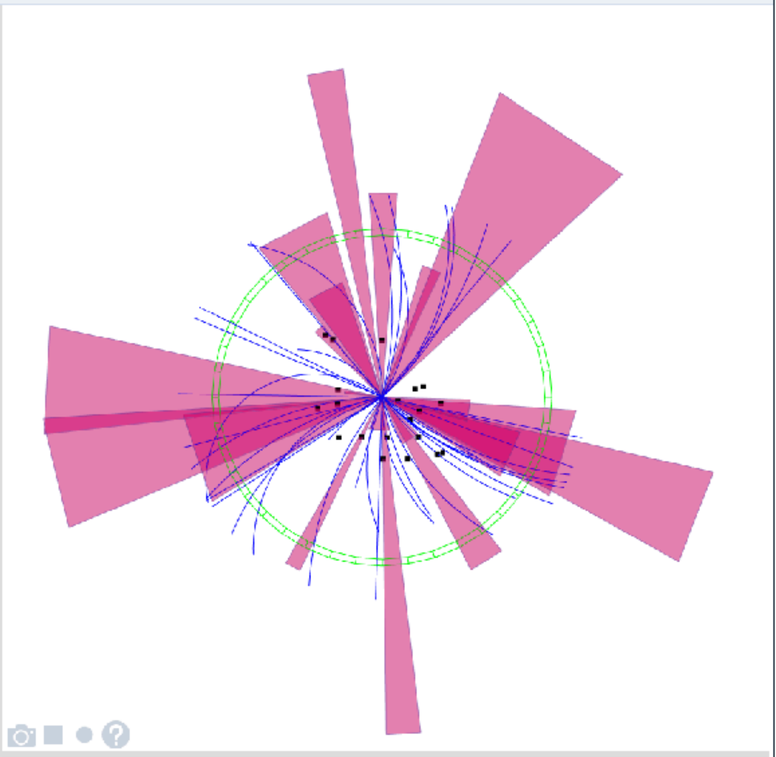


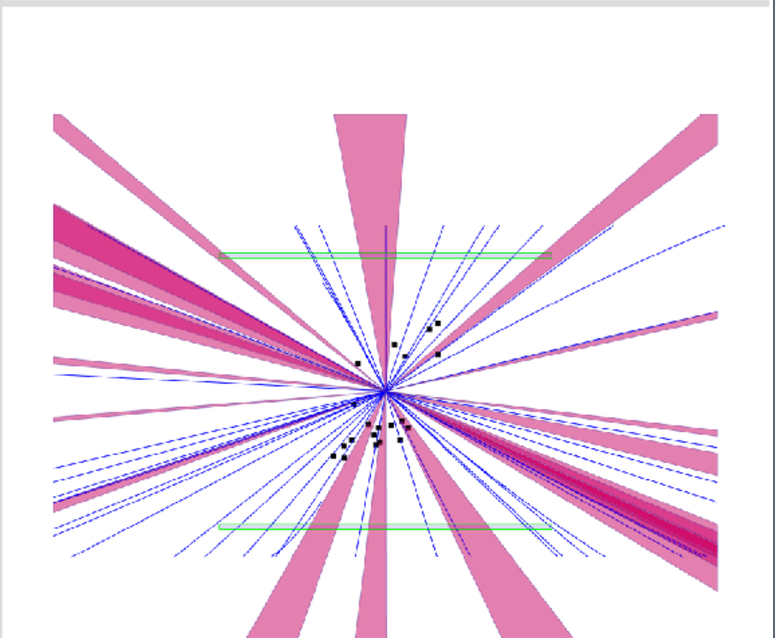
- EveWorld
  - Viewers
    - Default Viewer
    - RPhi View
    - RhoZ View
  - Scenes
    - Geometry scene
      - Barrel 1
    - Event scene
      - Hits
        - Points_1
      - Tracks
      - Jets
    - RPhi Geometry
    - RPhi Event Data
    - RhoZ Geometry
    - RhoZ Event Data
  - EventManager

Points_1 (ROOT::Experimental::TEvePo...

RnrSelf	☑
MarkerSize	12
MarkerXY	4.27327632904053







RForest

- Answer challenge "XYZ and simplistic data model is better than ROOT"!
  - be faster than everyone else, even for simple cases
  - optimize on current and remove anachronistic use cases
- Better usability by type-safe interfaces, separation of concerns (data model, serialization, backend storage, etc)
- Prototype for struct of vectors<primitive>, e.g. CMS's nanoAOD

RDataFrame / TMVA / PyROOT / Parallelization

- TMVA: ROOT data into machine learning
- RDataFrame for writing highly efficient TTree-/RForest-based analyses in a simple yet composable way
- PyROOT as real analysis frontend alternative with ROOT, for quick coding and highly efficient data transfer
- Combined: lego blocks used to build an analysis

RDataFrame / TMVA / PyROOT / Parallelization

- For highly efficient TTree/RForest analyses in a simple yet composable way
- Uses all your (for now: local) cores
- Simple: compact, modular, declarative code. Don't bother with reading data, iteration etc; +/- same code for C++ in Python
- Robust: type safety lets us complain if code does not match data
- Published (i.e. out of ROOT::Experimental) in June's production **release v6.14**

Enrico: RDataFrame

RDataFrame / **TMVA** / PyROOT / Parallelization

- Better numpy integration
- Cross validation, in parallel (multi-processing)!
- Adapters to external backends: TensorFlow, Keras, scikitlearn; for training *and* evaluation or only training (and evaluation in TMVA proper)
  - inference benefits from TMVA's knowledge of ROOT I/O
- New ML tools in TMVA: CNN (v6.12), RNN (v6.14), work ongoing for GAN
  - using CPU (openBLAS) or GPU (CUDA)

Kim: TMVA

RDataFrame / TMVA / **PyROOT** / Parallelization

- Provide support!

Enric: PyROOT

- Better numpy integration
- Re-basing PyROOT on top of cppyy, Wim Lavrijsen's "fork" of PyROOT, sharing development and goals: cmake -Dpyroot_experimental=On
  - C++ lambdas, move semantics for efficient use of C++ through Python
  - extensible pythonizations for new C++ code, e.g. histograms
- Build for both Python 2/3 without having to rebuild ROOT

RDataFrame / TMVA / PyROOT / **Parallelization**

- Parallelism everywhere:
  - I/O, e.g. tree merging across threads
  - fitting, e.g. vectorized evaluation and parallelized minimization
- VecCore as abstraction for vectorized operations
- RooFit is back on track, with team around Wouter Verkerke plus new CERN fellow starting October: support and development
- Continuous performance monitoring

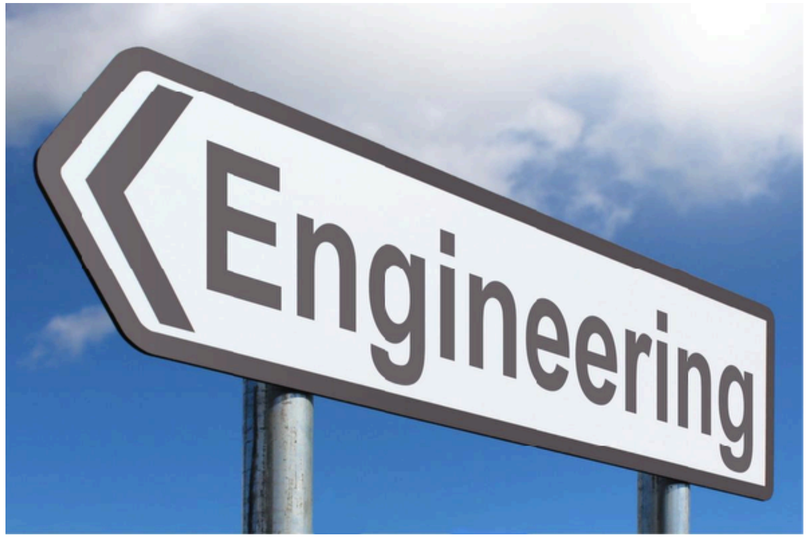
Lorenzo: Vectorization
Danilo: Future Parallel
Philippe: R/W Lock
Guilherme: BufferMerger

Lorenzo: Fitting/Modeling

Oksana: ROOTBench

FOUR GIANT DEV AREAS for that small a team?

- Prototype-driven development over years
  - ROOT6 taught us not to oversell nor underestimate work, but that we can trust us and the community
- Building on that experience plus early exposure
  - learned how to trigger early feedback: worked better for RDataFrame than before for new histograms
- Needs long-term team members from multiple affiliations



In Other News

- LZ4 new default compression algorithm v6.14
  - fast fast fast for reading (analysis), try it out!
  - back-ported read support to most old branches: pick up their patch release!
- cling gets a CUDA backend
- R&D on ROOT as build-on-demand tool
- ROOT works with C++ modules as dictionary

ROOT Workshop!

Oksana: root-get
Yuka: C++ Modules

Conclusion

ROOT: Back to the Future

- 1990s, ROOT started by needing to prove itself against alternatives
  - we are back in that situation, and we accept the challenge!
- Delivering a simpler, friendlier, more robust ROOT
  - address the real issues of physicists in a relevant and applicable way, instead of a series of blue-sky developments
  - yes R&D, but guided by prototypes and early feedback

World Domination, Final Words, Acknowledgements

- TMVA, cling and soon more help us conquer the world, despite HENP target
- We sincerely enjoy delivering high quality, production grade features that have an impact, and we will continue to do that, see numerous CHEP contributions
- Only possible due to CERN, Fermilab, DIANA-HEP, Google (SoC), unnamed companies, and many physicists and volunteers
- And thanks to our users for reporting bugs, providing patches, helping out on the forum, for constructive criticism as well as for cheers and praise

ROOT

- https://root.cern
- https://cern.ch/forum
- https://github.com/root-project
-  @root-project
-  https://www.linkedin.com/groups/1826455
- root-dev@cern.ch

ROOT Users' Workshop:
https://cern.ch/root2018