Matthias Mozer, University of Wisconsin, Madison

Root experience in the CMS experiment

Disclaimer

- This is not an official CMS sanctioned talk
- Focused on personal experience in various analyses
- Gives the view of someone with limited coding skills
- Member of CMS for 6 years, worked on:
  - Z' search
  - Z+jets cross section measurements
  - H ZZ 2l2q search
  - Graviton ZZ 2l2q search
  - H ZZ 4lepton search

  => all using somewhat different analysis flow
  => May not be representative

Outline

- General CMS analysis workflows
- Focus topic: RooFit
- Focus topic: pyRoot
- Leftovers
- Conclusion

CMS Analysis Steps

CMS data („AOD“)

Enriched CMS data („PAT“)

Custom objects in CMSSW files

Heavy TTree

Filtered/light TTree

RooDataSet

- Start at std. CMS data
- Go over several processing steps to final results

- Number of steps differs by analysis
- Most commonly 2-3 steps

CMS Analysis Steps

CMS data („AOD“)

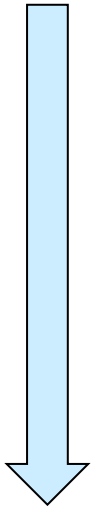
Enriched CMS data („PAT“)

Custom objects in CMSSW files

Heavy TTree

Filtered/light TTree

RooDataSet



Technically involves root files, but handled by CMSSW framework

Most resource intensive steps in here

CMS Analysis Steps

CMS data („AOD“)

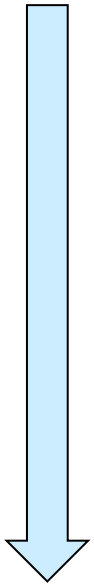
Enriched CMS data („PAT“)

Custom objects in CMSSW files

Heavy TTree

Filtered/light TTree

RooDataSet



Direct user-interaction

- Trees
- Histos
- Fits
- …

Why so many steps?

- Organic code growth
- Inheritance (not OO)
- Start with few steps
- Add additional variables / looser selection
- Output size grows unwieldy / processing gets slow
- Add additional step with more practical files
- Not a ROOT problem in itself
- Leads to lot of copy & paste
  - When adding variables
  - When adding additional step

Typical code

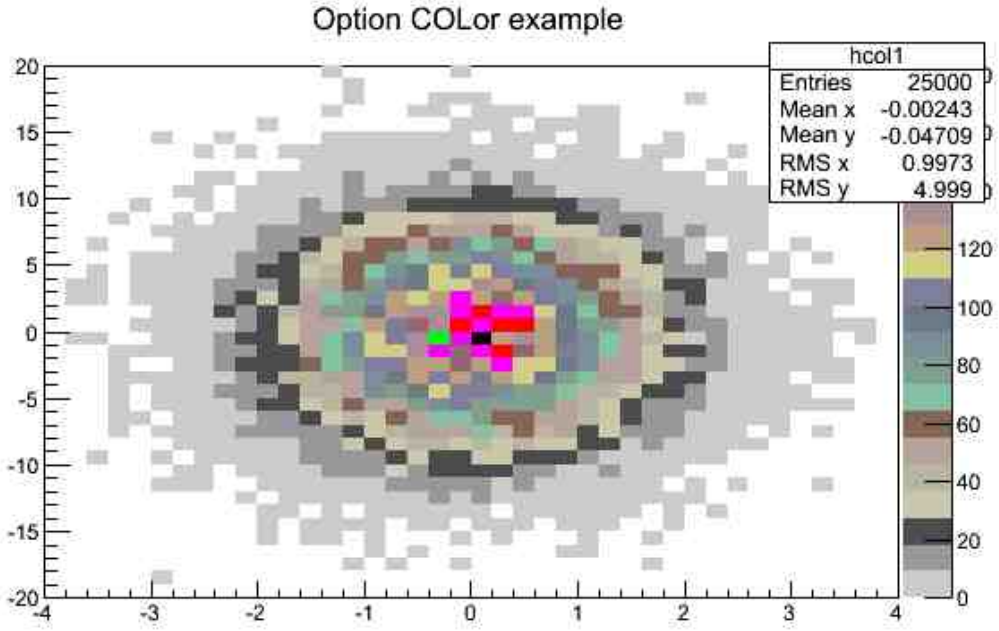
- Simple loops over TTrees
- MakeClass
- Advantages:
  - Quick to set up
  - Quick runtimes
  - Easy to share
  - Accessible to people with low programming skills (i.e. M.M.)
- Disadvantages
  - Common copy & pasting prone to errors
  - Can grow unwieldy for complex analysis

```
 newTree->Branch("bkg_m4l_ResDown",&bkg_m4l_ResDown
//pt/rapidity
newTree->Branch("p0_pt",&p0_pt,"p0_pt/F");  // mult
newTree->Branch("p0_y",&p0_y,"p0_y/F");  // multipl
newTree->Branch("bkg_pt",&bkg_pt,"bkg_pt/F");  // m
newTree->Branch("bkg_y",&bkg_y,"bkg_y/F");  // mult
newTree->Branch("VAKD",&VAKD,"VAKD/F");  // discrim

//interference weight
float interfWeight;
newTree->Branch("interfWeight",&interfWeight,"inter

for(int iEvt=0; iEvt<(max<0?sigTree->GetEntries():m

  if(iEvt>=sigTree->GetEntries()) break;

  if(iEvt%1000==1) {
    cout<<"---------\n event: "<<iEvt<<endl;
  }
  sigTree->GetEntry(iEvt);
```

Making Plots

- Current default settings much better than a few years ago
- Final plots reasonably nice
- Achieved by maintaining a (often customized) macro with a unified CMS style

- Adding/Aligning texts and legends still quite cumbersome
- Default palette for 2d plots still a disaster



CMS Analysis Summary

- Event by event work on ntuples still the bread & butter of physics analysis
- TTree interface is cumbersome but by now everyone is used to it.
- Visually appealing plots are not always easy, but decent recipes are available.
- C/C++ (still?) widely known by students, dominating at postdoc/prof. level

RooFit

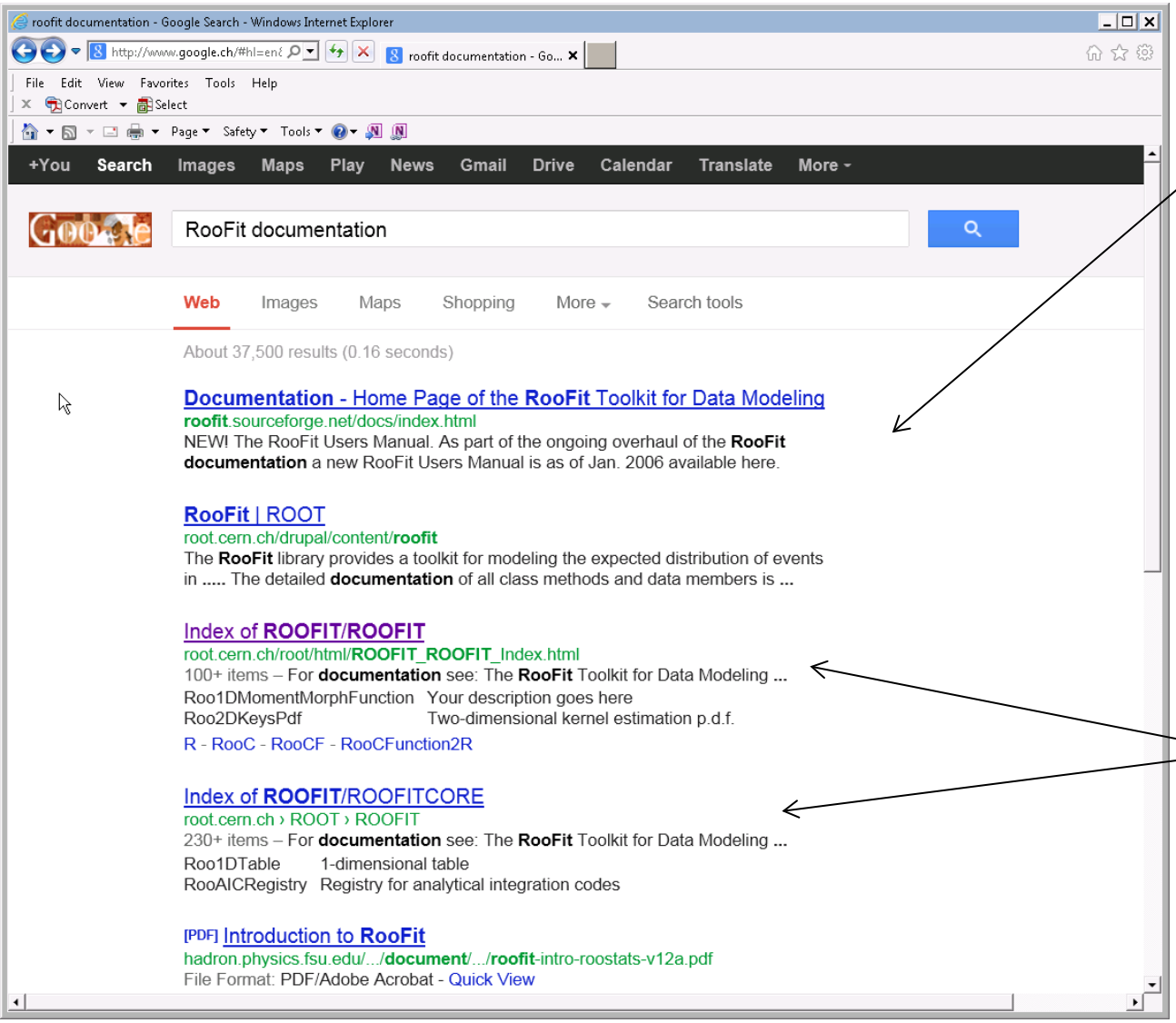
- RooFit is used for many CMS analysis
  - All Higgs analysis use RooFit via Higgs combination program
  - Availability of an „approved“ limit setting program means that almost all searches use it too
  - Additionally used for various cross section measurements via signal + background fits

=> Central to many important results
=> But not as smooth in everyday usage as baseline root

RooFit Issues

- Stability (seems much improved now)
  - Common updates (out of step with std CMS environment)
  - Weighted datasets
  - Error computation with weighted datasets
  - RooWorkspace would have been great to have around earlier
  - Interface changes silently breaking backwards compatibility
- Binning and ranges not always intuitive
- Combinations of binned and unbinned pdfs/dataset can be unintuitive
- Setting appropriate levels of output with RooMsgService can be difficult.
- Documentation could be improved

Documentation



From 2006, before Integration in root

Class docu. Only useful if already familar Can be terse

Documentation

roofit documentation - Google Search - Windows Internet Explorer

http://www.google.ch/#hl=en&

roofit documentation - Go...

File Edit View Favorites Tools Help

Convert Select

Page Safety Tools

+You Search Images Maps Play News Gmail Drive Calendar

Google RooFit documentation

Web Images Maps Shopping More Searc

About 37,500 results (0.16 seconds)

Documentation - Home Page of the RooFit Toolkit for
roofit sourceforge.net/docs/index.html
NEW! The RooFit Users Manual. As part of the ongoing overhaul c
documentation a new RooFit Users Manual is as of Jan. 2006 av

RooFit | ROOT
root.cern.ch/drupal/content/**roofit**
The **RooFit** library provides a toolkit for modeling the expected dis
in ..... The detailed **documentation** of all class methods and data

Index of ROOFIT/ROOFIT
root.cern.ch/root/html/**ROOFIT_ROOFIT**_Index.html
100+ items – For **documentation** see: The **RooFit** Toolkit for Data
Roo1DMomentMorphFunction Your description goes here
Roo2DKeysPdf Two-dimensional kernel estimation
R - RooC - RooCF - RooCFunction2R

Index of ROOFIT/ROOFITCORE
root.cern.ch › ROOT › ROOFIT
230+ items – For **documentation** see: The **RooFit** Toolkit for Data
Roo1DTable 1-dimensional table
RooAICRegistry Registry for analytical integration codes

[PDF] **Introduction to RooFit**
hadron.physics.fsu.edu/.../**document**/.../**roofit**-intro-roostats-v12a.pdf
File Format: PDF/Adobe Acrobat - Quick View

RooFit | ROOT - Windows Internet Explorer

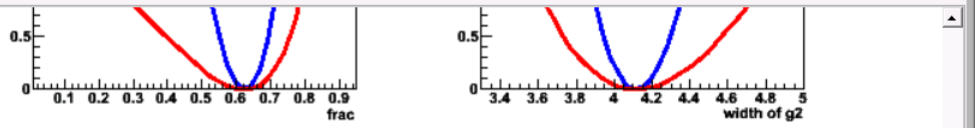
http://root.cern.ch/

RooFit | ROOT

File Edit View Favorites Tools Help

Convert Select

Page Safety Tools



Documentation

Quick Start Guide

A 24 page quick start guide for RooFit version 3.00 is available here. This guide was last updated in July 2009 and documents the workspace/factory concept in addition to brief introductions to various aspects of data modeling.

Users Manual

A 134-page users manual for RooFit version 2.91 is available here. This manual was last updated in October 2008, but is still completely valid. Another update that includes all new features added in version 3.00 will appear before October 2009

RooWorkspace missing
Factory interface missing
Outdated on weighted datasets

RooFit Summary

- RooFit successfully hides many of the difficult issues of maximum likelihood fits
  - Widespread use in CMS
- Initially quick update schedule was aggravating
  - Try to keep it stable
- Some unintuitive behavior remains
  - Not necessarily actual bugs
  - Often just shortcomings in documentation

pyRoot

- Not used as widely in CMS as C++-root
- Some analysis completely based on pyRoot (very few)
- Tap growing python knowledge in student base
  - But can be challenging already for people from my age group
- Good synergy with python based configuration of CMSSW software framework

pyRoot Advantages

- pyRoot shines were it uses features of python that are not as easily available in C++
- Easy tree access:

```
#read the tree
infile = root.TFile.Open(filename)
tree = infile.Get("SelectedCandidates")
for event in tree:
    for i in range(event.nCands):
        if event.nXjets[i]==njet and event.region[i]==1 \
        and event.mZZ[i]> mzz.getMin() and event.mZZ[i]< mzz.getMax() :
```

  - Less copy&paste => less error prone
- Easier non-physics programming
  - String parsing / composition
  - File system interactions
- Easier to integrate scripting of analysis with actual analysis
  - Command line parameters
  - File system interactions

pyROOT Shortcomings

- pyRoot suffers where it's just a basic translation of C++
- Call by reference

So, the types are ROOT.Long and ROOT.Double which pass through long& and double&. Something like:

CODE: SELECT ALL

```
d1, d2 = ROOT.Long(0), ROOT.Long(0)
myGraph.GetPoint( 41, d1, d2 )
```

should do the trick.

- Returned pointers

CODE: SELECT ALL

```
y_buff = tgraph.GetY()
N = tgraph.GetN()
y_arr = numpy.ndarray(N,'d',y_buff)
```

- Reserved keywords

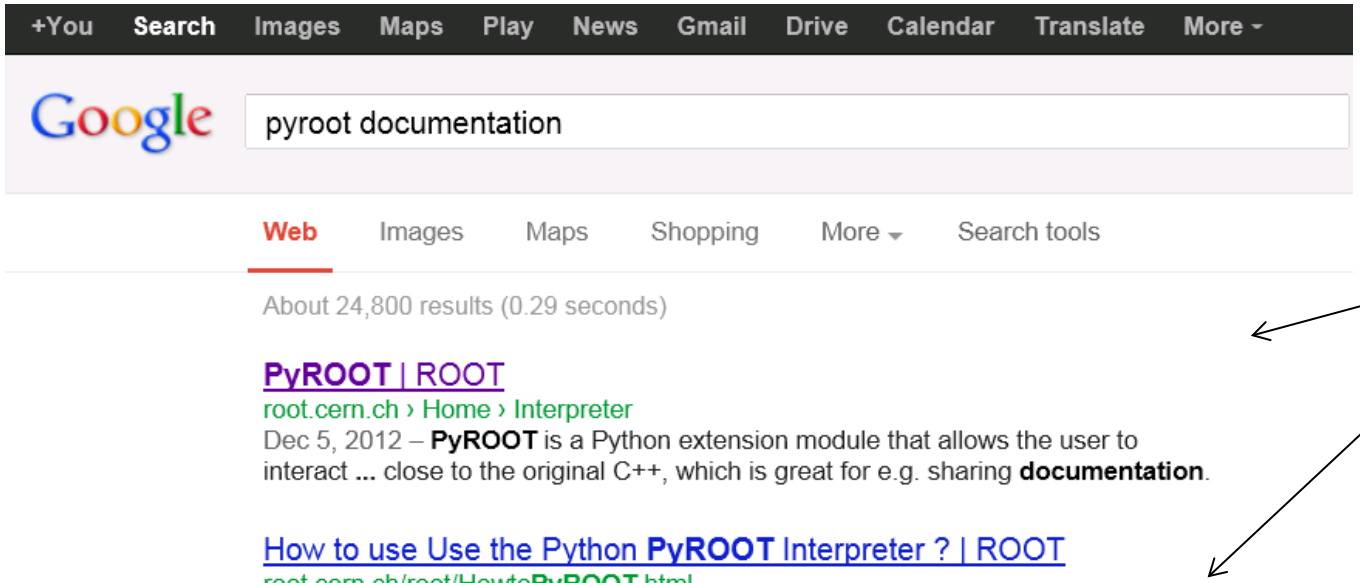
yes, "import" is a keyword and can thus not be used as a method. Easiest workaround would be:

CODE: SELECT ALL

```
# Import model and all its components into the workspace
getattr(w,'import')(model)
```

- Poor Integration of Titerator
  - Few things natively iterable

Documentation



Almost identical

Technical doc.

Same document as above

Documentation

+You Search Images Maps Play News Gmail Drive Calendar Translate More

Google pyroot documentation

Documentation from 2006 / root 5.12
Still up to date, who knows?

root.cern.ch › Home › Interprete
Dec 5, 2012 – **PyROOT** is a Pyt
interact ... close to the original C

How to use Use the Pytho
root.cern.ch/root/HowtoPyROO
Documentation ... **PyROOT** is
interact with any ROOT class fr
PyROOT offers the possibility tc
start ...

PyROOT
root.cern.ch › ROOT › BINDING
Sep 20, 2012 – namespace **Pyl**
documentation or ROOT in ge

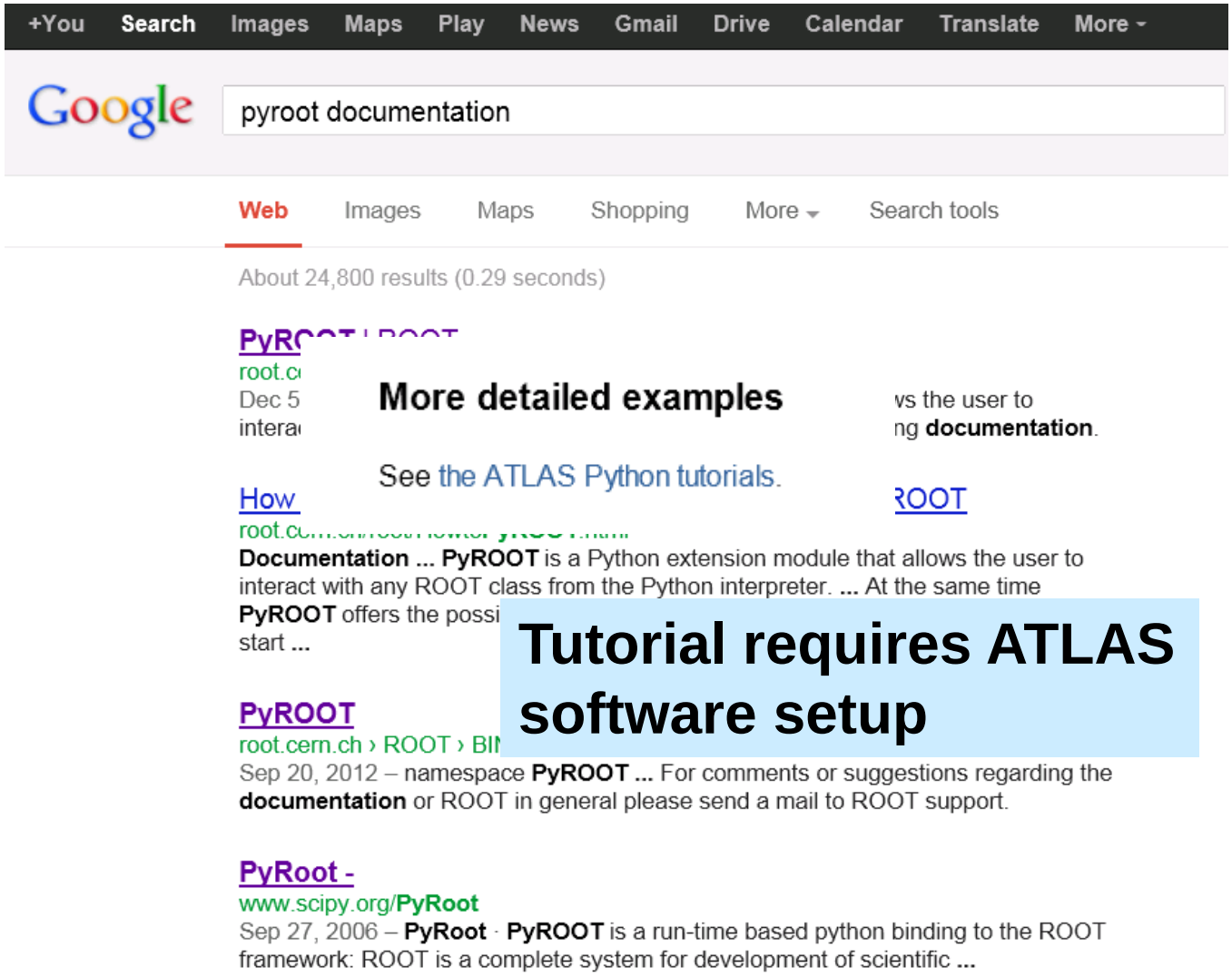
PyRoot -
www.scipy.org/PyRoot
Sep 27, 2006 – **PyRoot** · **PyROOT** is a run-time based python binding to the ROOT framework: ROOT is a complete system for development of scientific ...

Revision History		
Revision 1.1.1	October 2008	wlav
Minor fixes in the text.		
Revision 1.1	June 2006	wlav
Updates for ROOT production release 5.12.		
Revision 0.1 -- 1.0	June 2004 -- December 2005	wlav
Details removed.		

Abstract

Note
The document describes PyROOT as shipped with ROOT production release 5.12.

Documentation



Tutorial requires ATLAS software setup

PyRoot Summary

- Python is well known among younger students
  - Get students work productively more quickly
- pyRoot avoids some of the cumbersome constructs of C++ root (and C++ in general)
  - Less copy & paste
  - Fewer opportunities for coding mistakes
- Documentation could use some improvement
- Better integration of python language features
  - Available in rootpy
    not tried yet myself
    requires private installs, limiting portability of code

Parallelism

- Simple batch submission the preferred tool for parallelism.
- RooFit multi-core support very useful
- PROOF not used in any analysis I‘ve seen.

- Why is that?
  - PROOF doesn’t appear to be very widely deployed
  - Interactive work on large samples with heavy processing is rare (cause or effect?)
  - Reliance on TSelector, which I’ve rarely seen being used otherwise

  => advertisement / user education seems to be a major issue

LHE files

- Well established in theory community (Many MC generators produce events in LHE files)
- Basic xml implementation of old FORTRAN accord
- Not natively read by root
- A multitude of cobbled together LHE-parsers

hep-ph/0609017
CERN-LCGAPP-2006-03
September 2006

A standard format for Les Houches Event Files

J. Alwall [a] A. Ballestrero [f] P. Bartalini [v] S. Belov [h] E. Boos [i]
A. Buckley [p] J.M. Butterworth [q] L. Dudko [i] S. Frixione [m,e]
L. Garren [s] S. Gieseke [c] A. Gusev [j] I. Hinchliffe [t] J. Huston [u]
B. Kersevan [k] F. Krauss [b] N. Lavesson [ℓ] L. Lönnblad [ℓ]
E. Maina [f] F. Maltoni [a] M.L. Mangano [m] F. Moortgat [n]
S. Mrenna [s] C.G. Papadopoulos [d] R. Pittau [f] P. Richardson [p]
M.H. Seymour [m,r] A. Sherstnev [o] T. Sjöstrand [m,ℓ,*] P. Skands [s]
S.R. Slabospitsky [j] Z. Wąs [g] B.R. Webber [o] M. Worek [g]
D. Zeppenfeld [c]

Conclusion

- Overall good experience with ROOT
  - Definitely progress from PAW
- Single biggest productivity boost likely from improved documentation / user education / outreach:
  - Many useful tools not widely known
  - Better advertisement of new features
  - Is there a technical writer on the ROOT team?
- Many smaller issues can be improved, but:
  - We've worked for so long with root that we don't notice many of the small annoyances any more
  - Typical users aren't necessarily good programmers
    => advanced C++ constructs may be counterproductive
  - Many patterns of root usage are deeply ingrained & physicists rather lazy
    => changes should provide compelling improvements