Moving ROOT Forward

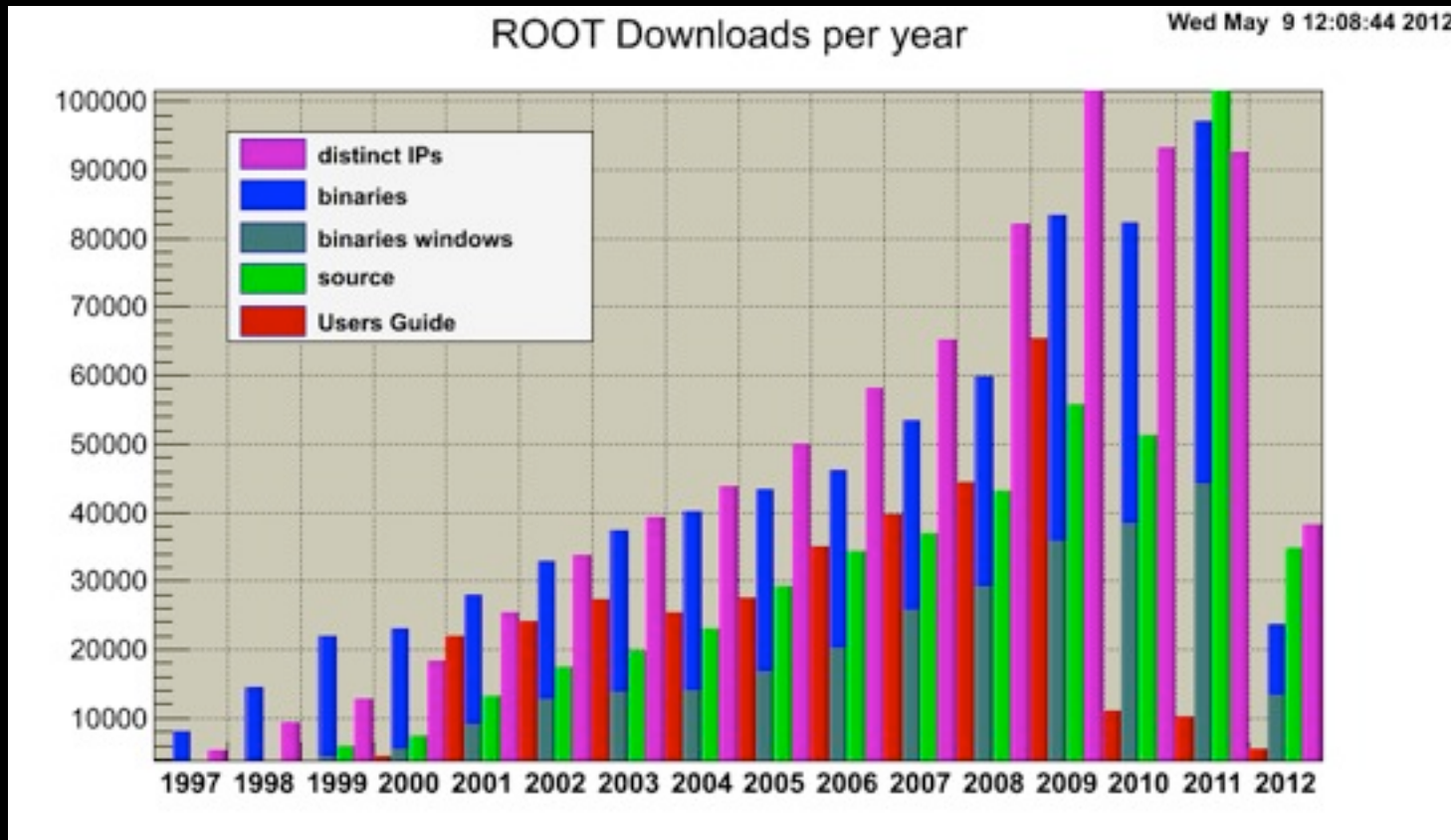## Balancing between stability and innovation

Fons Rademakers
CERN PH/SFT

CHEP'12, 23-May, 2012, NYC, US.

ROOT Is Doing Great

- Ever increasing number of users
  - 6100 forum members, 62000 posts, 1300 mailing list members
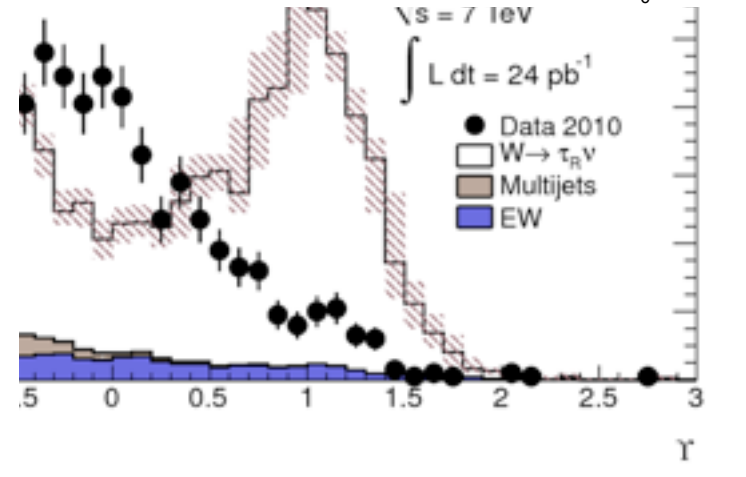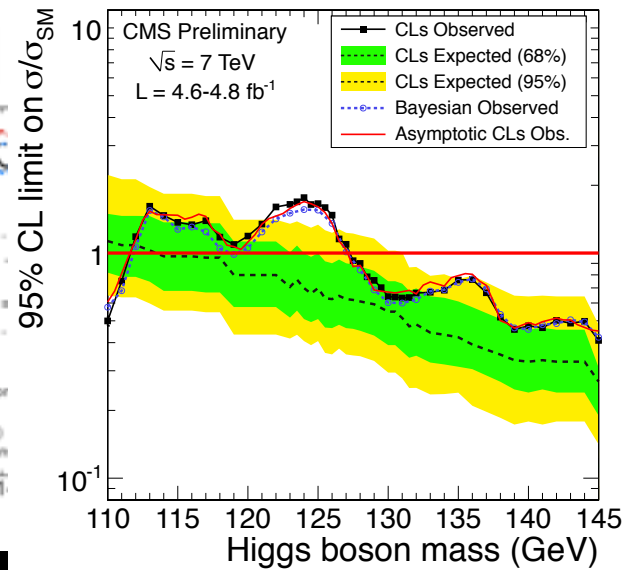 - Used by basically all HEP experiments and beyond
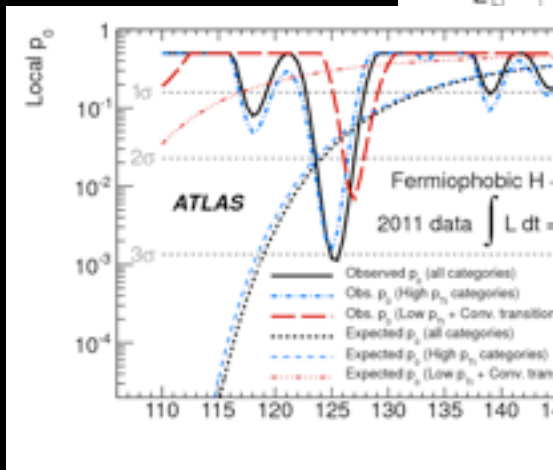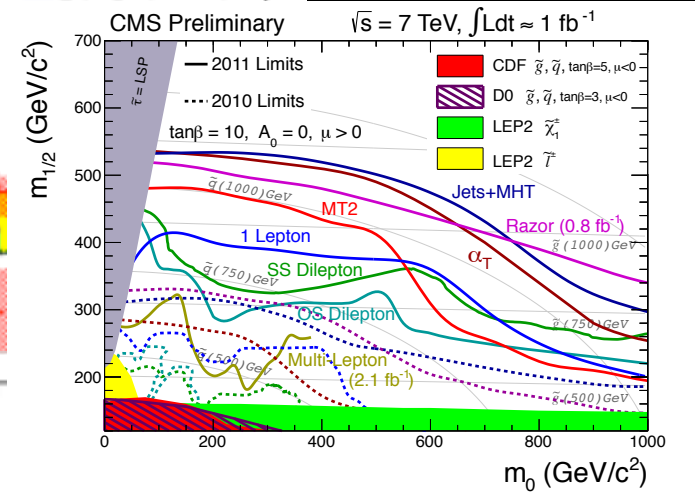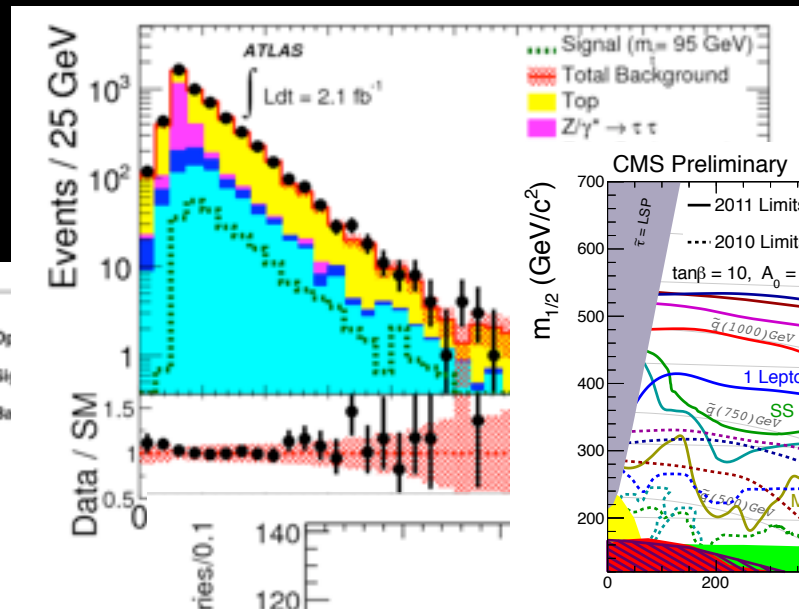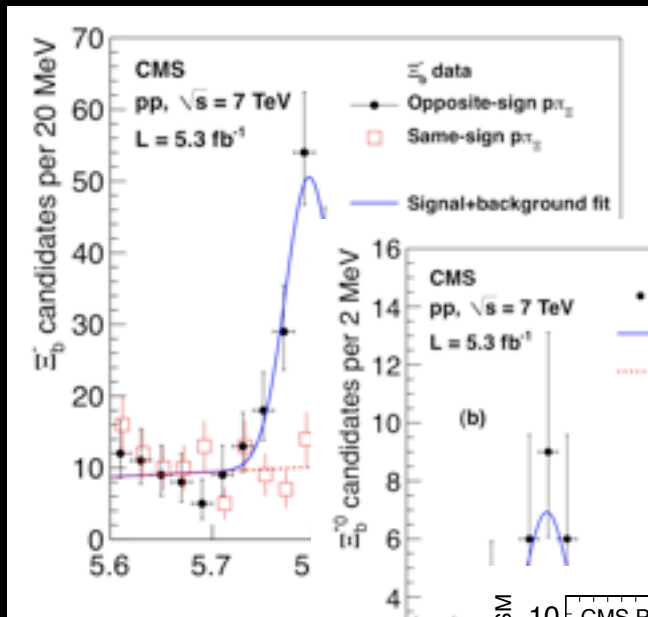
The HEP Data Format

As of today
119 PB
of LHC data
stored in ROOT format

ALICE: 23.5PB, ATLAS: 35PB, CMS: 55PB, LHCb: 5.2PB

The Standard for Graphs and Plots

The Standard for Graphs and Plots

New Interpreter

- Replacing CINT by new cling: correctness, full C++

- Cling based on LLVM and clang compiler libraries
  - "New" open source compiler suite, default on OSX Lion

- Compiler-grade C++ on its way to C++11

- Best diagnostics on the market:

CINT

```
Error: Can't call map<string,const char*,less<string>,allocator<const
string,const char*> > >::operator[]((char*)0x255a9e8) in current scope
err.C:19:
Possible candidates are...
(in map<string,const char*,less<string>,allocator<const string,const
char*> > >)
Error: improper lvalue err.C:19
```

```
err.C:19:15: error: assigning to 'mapped_type' (aka 'const char *') from
      incompatible type 'double'
   myMap["A"] = 12.3;
              ^ ~~~~
```

**cling**

New Interpreter, continued

- Cling released in July 2011
  - Fully functional C and C++ interpreter
  - Just-in-Time compilation, i.e. always ACLiC
  - Still a few issues to solve, e.g. reloading of code
  - Support multiple interpreter objects (c.f. multi-threaded context)
- Integration with ROOT ongoing
  - Interfaces via TCling and TClass
  - Precompiled headers as dictionary, no huge source files
- See the cling presentation on Monday @ 1:30 pm
  - http://indico.cern.ch/contributionDisplay.py?contribId=255&confId=149557

Improvements in I/O

- Parallel Tree merging (see next slides)
- Asynchronous prefetching
  - When enabled, a separate thread will fetch the TTreeCache blocks that are likely to be needed next and the main thread continues the normal data processing
- Automatic support for more than one TTreeCache per file.
  - TTree::SetCacheSize(Long64_t) no longer overrides nor deletes the existing cache. Each cache is completely independent
- Support for HDFS, Amazon S3, CloudFront and Google Storage
- Support for LZMA compression

Parallel Merging

- Existing solution:
  - Processing and storing the partial output on each local node and then only at the end of the process, upload to the server and only when all slaves are done, read all files on the server and write to a single output file.
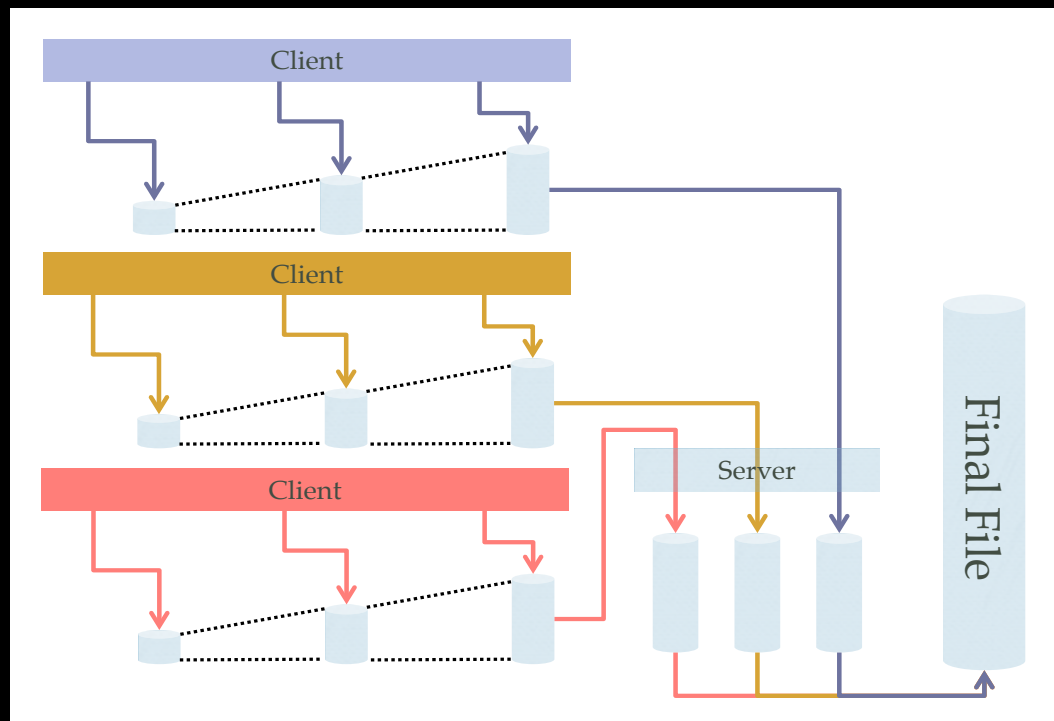
Parallel Merging

- New TFile implementations:
  - TMemFile: a completely in-memory version of TFile.
  - TParallelMergingFile: a TMemFile that on a call to Write will upload its content and reset the TTree objects.

- New solution:
  - Increase parallelism by having the slaves start uploading the TTree clusters directly to the server which immediately starts saving them in the final output file.
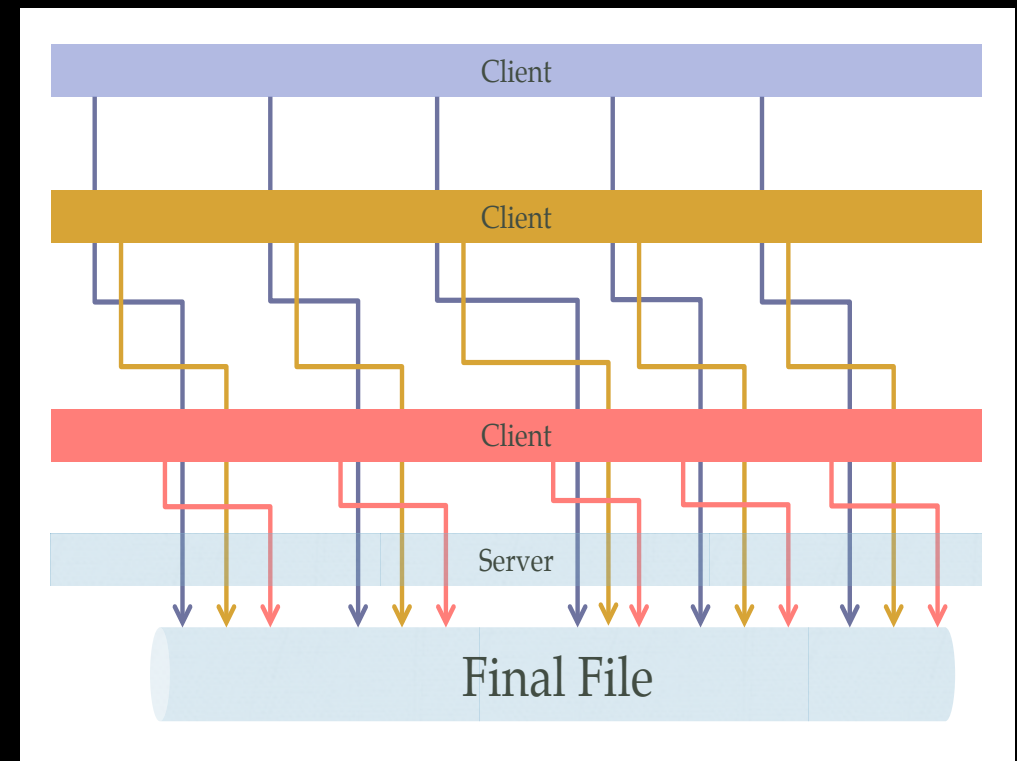
On-going I/O Development

- Continuous review of I/O performance in close collaboration with experiment experts

- Reimplementation of OptimizeBaskets

- Explore changing the on-file byte format to little endian

- Extension of the parallel merger

- Improvement of thread safety

- I/O Customization: Nested Objects

Parallelism for I/O

- 3 approaches:
  - Parallel Merge via external process
  - Support for one TFile per thread
    - Currently requires meta data (TClass, TStreamerInfo) to be fully build before starting parallel operation.
    - Planning to lifting this restriction in ROOT 6 (requires cling).
  - Internal use of spare cores
    - Ability to read multiple TBranch data in parallel
    - Top level branches can be uncompressed and un-streamed independently.
    - Prefetching of remote file content as a separate thread.

Improvements in Histogram Classes

- New class for binning multi-dimensional data using a kd tree (TKDTreeBinning)

  - Automatic binning of data

    - every bin will have same entries or same weight (content)

  - Efficient multi-dimensional histogram

  - Can be used to compare MC and data (make bins with MC and compare with the data)

  - Could be used for non-parametric density estimation using kernels
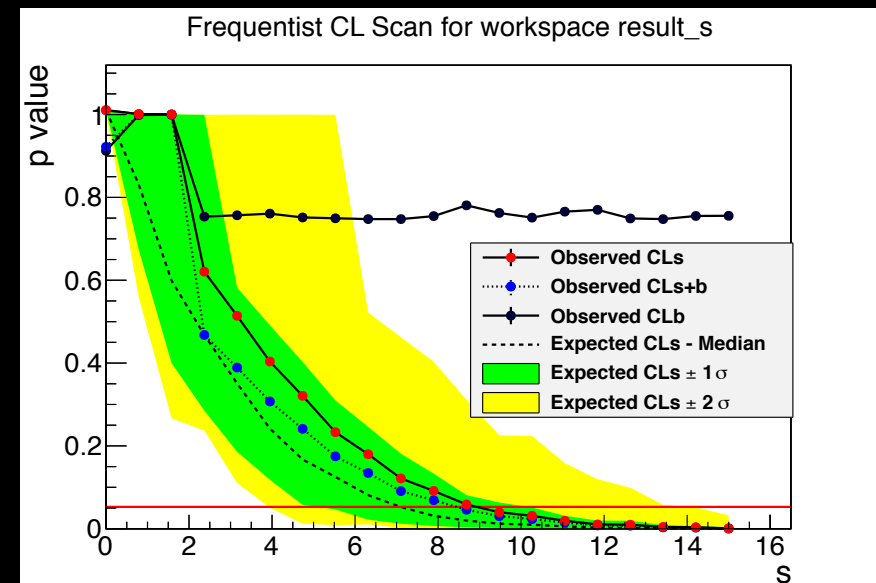

KDTree binning
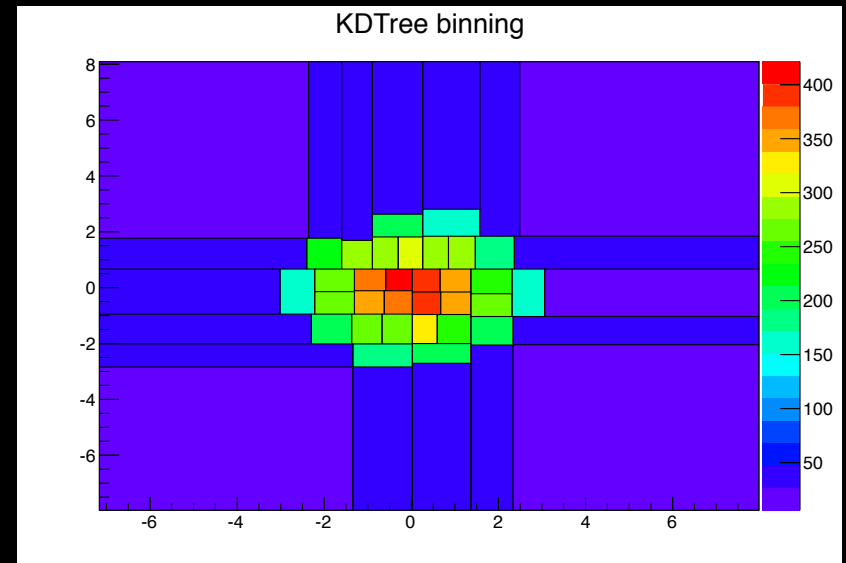
- Also new THn class

  - Multidimensional histograms with equal bins

  - For non-sparse multi-dimensional data

RooFit/RooStats

- Framework for statistical calculations
  - Standard tool for producing physics results at LHC
    - Parameter estimation (fitting)
    - Interval estimation (e.g limit results for new particle searches)
    - Discovery significance (quantifying excess of events)
  - Implement several statistical methods (Bayesian, Frequentist, Asymptotic)
  - New tools for model creation and combinations
    - Histfactory: make RooFit models (RooWorkspace) from input histograms
  - See the presentation on Thursday @ 1:30 pm
    - https://indico.cern.ch/contributionDisplay.py?sessionId=6&contribId=371&confId=149557

New RooStats Features

- Classes for computing exclusion limits and discovery significance using the procedure endorsed and agreed by ATLAS and CMS (CL_s method for limits)
  - Using both pseudo-experiments or asymptotic formulae
  - Compute observed limits, expected limits and bands

- Can be very time consuming, many pseudo-experiment need to be generated and fitted

- Working on improving performances
  - More efficient fitting for multi-parameter problems (improve the new version of Minuit)
  - Investigate parallelization techniques (PROOF, Multi-threads or porting on GPU)

Frequentist CL Scan for workspace result_s

Legend:
- Observed CLs
- Observed CLs+b
- Observed CLb
- Expected CLs - Median
- Expected CLs ± 1σ
- Expected CLs ± 2σ

Example: Higgs Combination

- Higgs combination results of ATLAS and CMS
  - Make combined model with several channels and more than 200 parameters
  - RooStats used successfully for producing results:
    - Could use different statistical methods for comparison
    - Required a lot of CPU for fitting each pseudo-experiment
      - Use Grid for submitting all jobs
   - On-going work on improving fitting performances
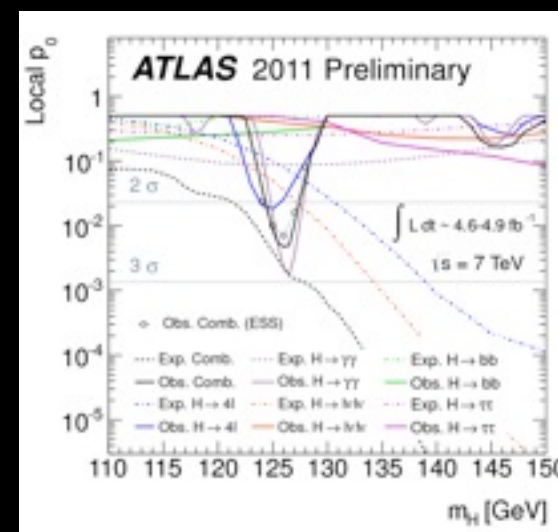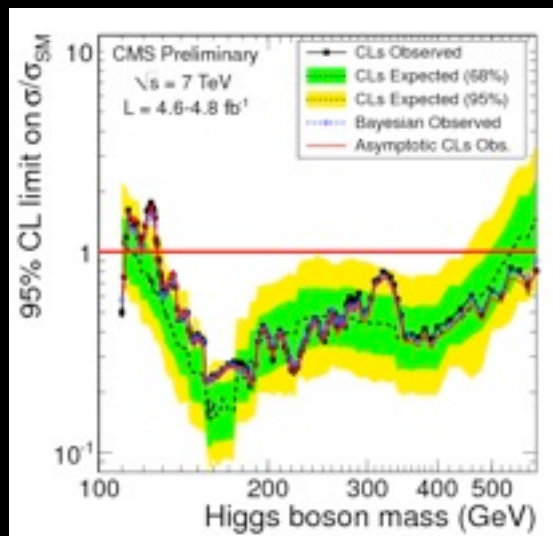
New PROOF Features

- Increasing number of sites and users deploying PROOF

- Deployment on ad-hoc PROOF clusters is made easy by PoD
  - See PoD talk on Monday @ 5:25 pm
    - http://indico.cern.ch/contributionDisplay.py?contribId=22&confId=149557

- The setup of an dedicated PROOF cluster including file management and file access using xrootd has been made easy by PEAC
  - See PEAC poster
    - http://indico.cern.ch/contributionDisplay.py?contribId=169&confId=149557

- An extensive PROOF bench suite has been developed allowing to test many aspects of PROOF related parameters (network, disk, cpu, etc.)

- Many stability improvements

PROOF Developments

- Improvement of the connection layer and of the work distribution engine (packetizer)

- Result merging optimization (see also parallel tree merging in I/O)

- Improvement of memory management (in merging of results, in sharing common data)

- Support for dynamic addition of workers

- Towards a fully-functional multi-master setup
  - Dataset management across clusters
  - Dynamic load-balancing across sub-masters
  - Dynamic re-organization of a set of workers in multi-tier structures to scale up to O(1000) cores
  - Needed for extreme platforms like IBM's BG/Q

New MacOS X Native Graphics Backend

- Uses native OSX Cocoa and Quartz for graphics
  - Supports easily new features like transparency, gradients and shadows
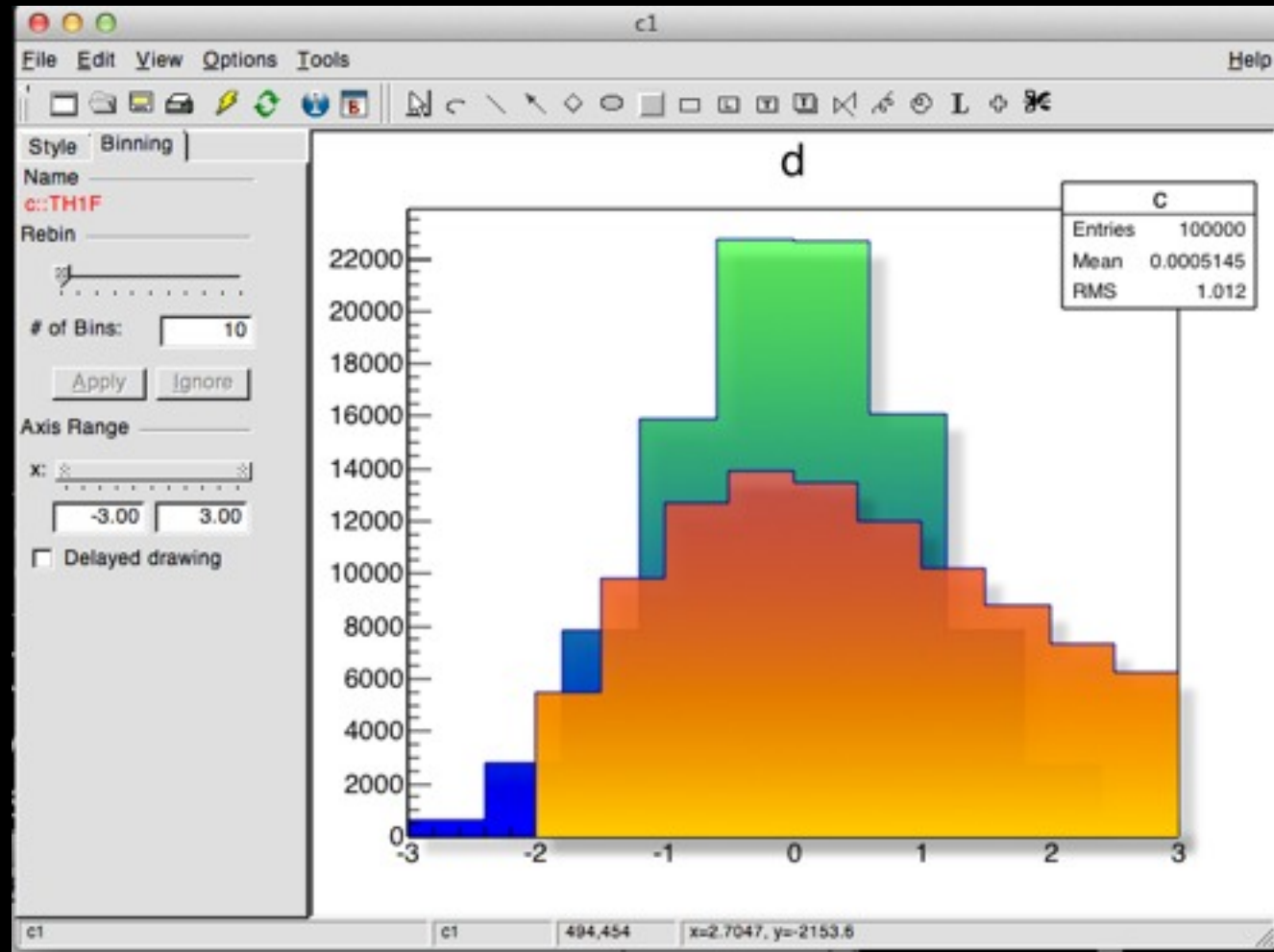
Transparency

- In parallel to OSX Cocoa and Quartz, transparency is also supported in PDF and TImage outputs.
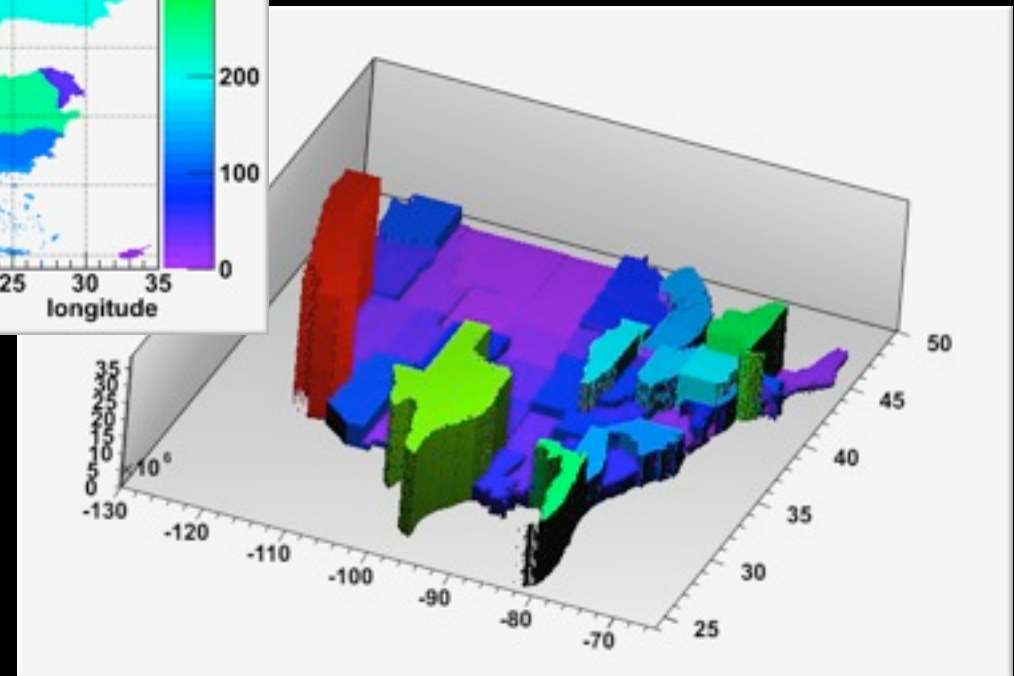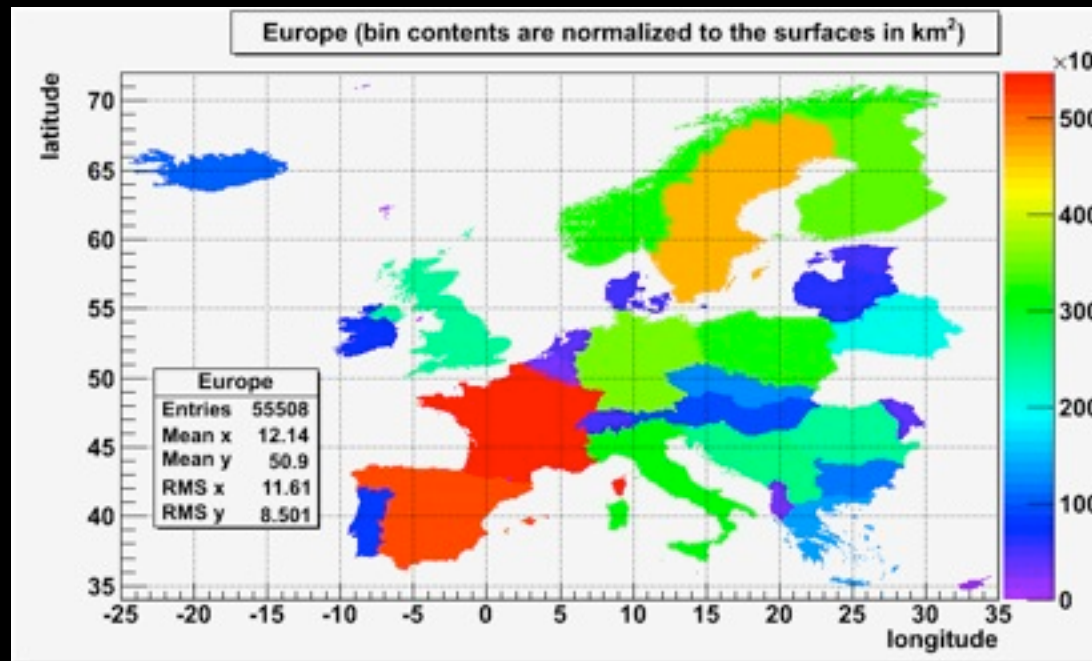
This provides new possibilities to visualize large data set. As shown on this parallel coordinates plots.

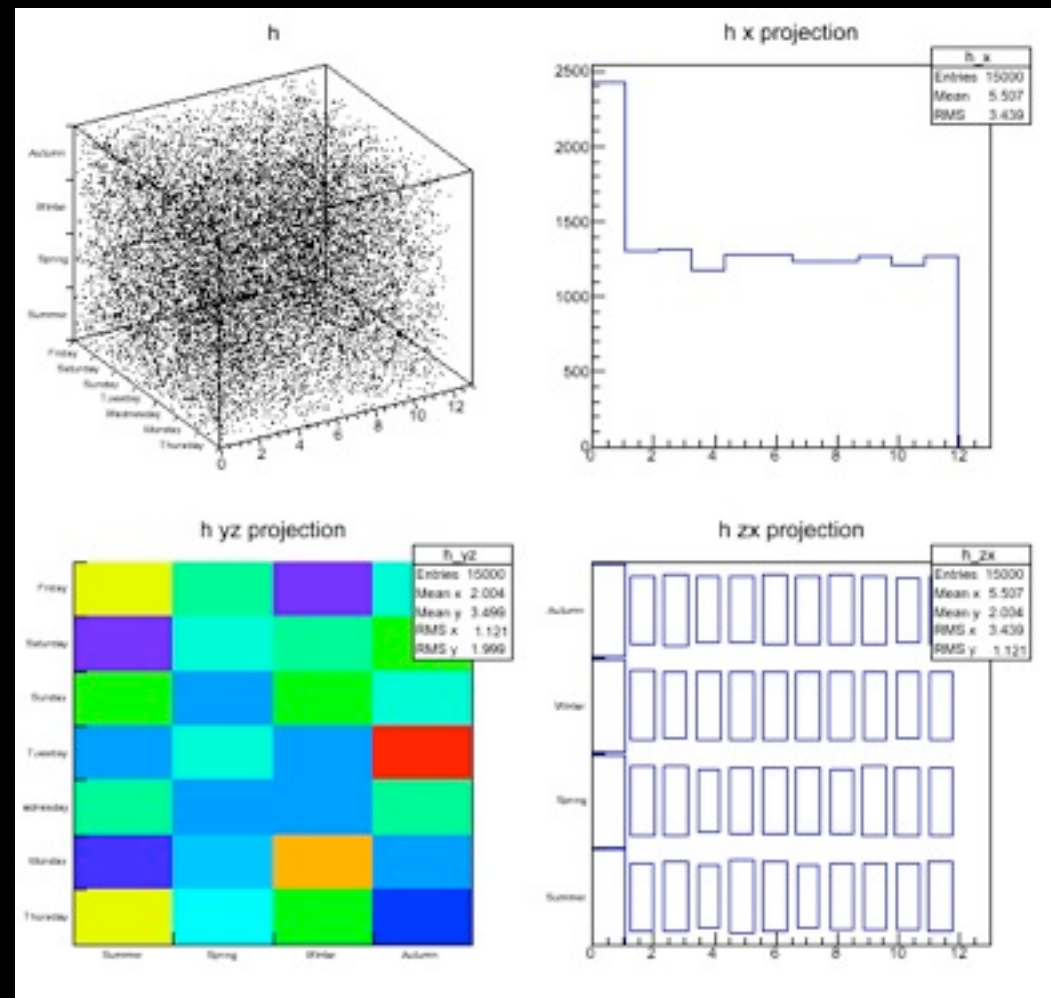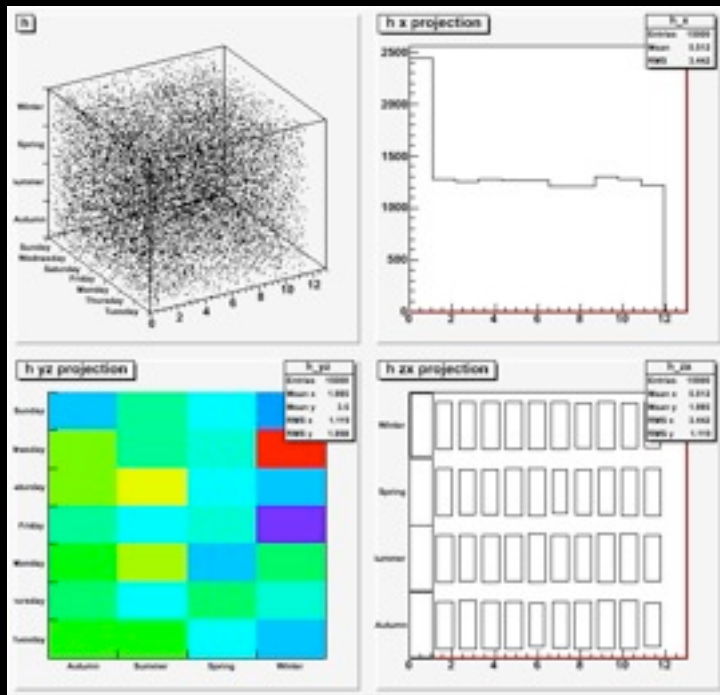These two plots are the same. On top without transparency: useless. Bottom with transparency: all the clusters clearly appear.
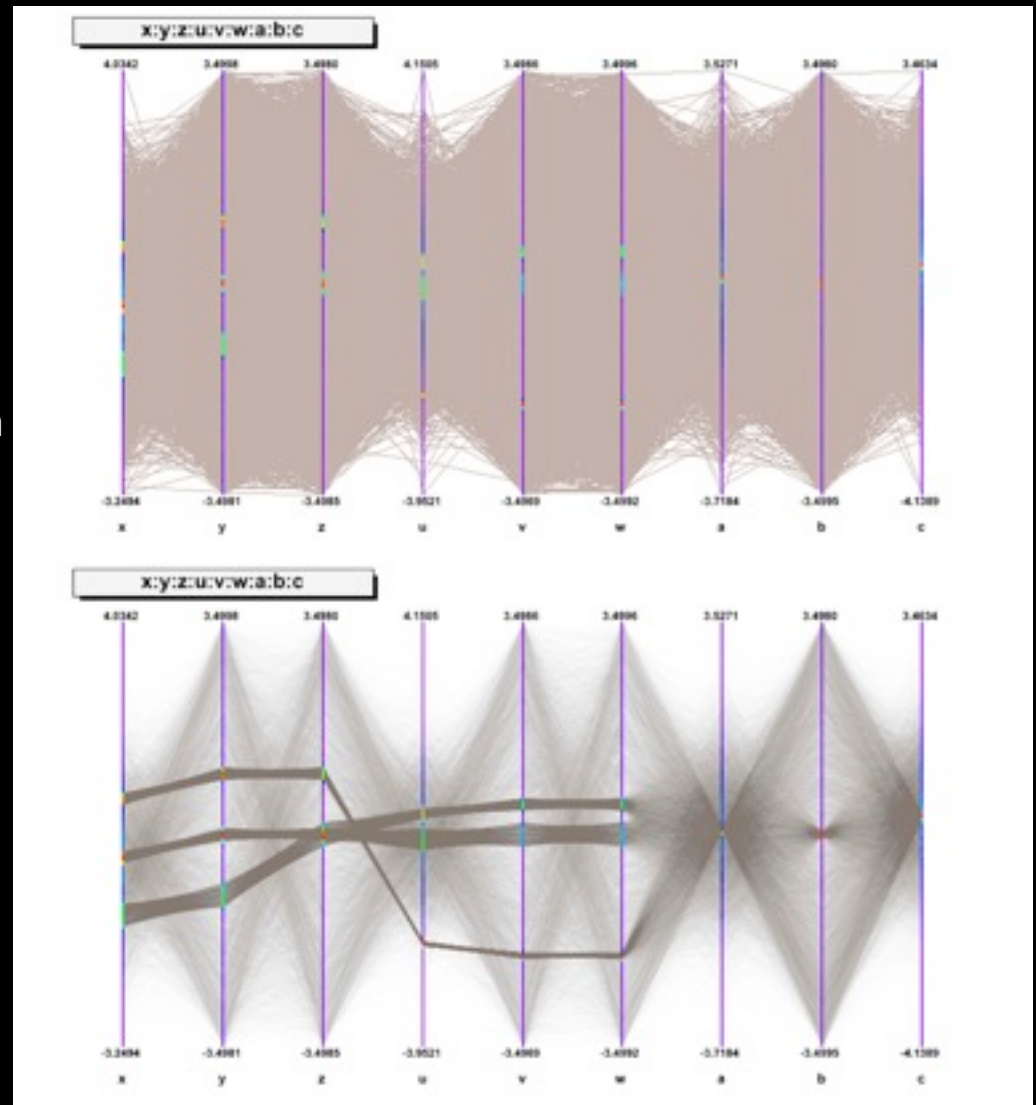
TH2Poly

- TH2Poly is a 2D Histogram class, inheriting from TH2, allowing to define polygonal bins of arbitrary shape.

"Refreshed" Default Graphics Style

- The new style "Modern" is a synthesis of different custom styles used by several major experiments.

New TeX Engine

$$\Pi_{j\geq 0}\left(\sum_{k\geq 0} a_{jk} z^k\right) = \sum_{n\geq 0} z^n \left(\sum_{k_0,k_1\cdots \geq 0} a_{0k_0} a_{1k_1}\cdots\right)$$

$$W^{3\beta}_{\delta_1\rho_1\sigma_2} = U^{3\beta}_{\delta_1\rho_1\sigma_2} + \frac{1}{8\pi^2}\int_{\alpha_1}^{\alpha_2} d\alpha'_2 \left[\frac{U^{2\beta}_{\delta_1\rho_1} -\alpha'_2 U^{1\beta}_{\rho_1\sigma_2}}{U^{0\beta}_{\rho_1\sigma_2}}\right]$$

$$d\Gamma = \frac{1}{2m_A}\left(\Pi_f \frac{d^3 p_f}{(2\pi)^3 2E_f}\right)\left| M(m_A -\{p_f\})\right|^2 (2\pi)^4 \delta^{(4)}\left(p_A -\sum_f p_f\right)$$

$$4\,\mathrm{Re}\left\{\frac{2}{1-\Delta\alpha}\chi(s)\left[\hat{g}_v^e \hat{g}_v^f (1+\cos^2\theta) + 2\hat{g}_a^e \hat{g}_a^f \cos\theta)\right]\right\}$$

$$p(n) = \frac{1}{\pi\sqrt{2}}\sum_{k=1}^{\infty}\sqrt{k}\,A_k(n)\frac{d}{dn}\frac{\sinh\left\{\frac{\pi}{k}\sqrt{\frac{2}{3}}\sqrt{n -\frac{1}{24}}\right\}}{\sqrt{n -\frac{1}{24}}}$$

New TeX Engine

Current

$$\Pi_{j\geq 0}\left(\sum_{k\geq 0} a_{jk}z^k\right) = \sum_{n\geq 0} z^n\left(\sum_{k_0,k_1,\ldots\geq 0} a_{0k_0}a_{1k_1}\cdots\right)$$

$$W^{3\beta}_{\delta_1\rho_1\sigma_2} = U^{3\beta}_{\delta_1\rho_1\sigma_2} + \frac{1}{8\pi^2}\int_{\alpha_1}^{\alpha_2} d\alpha'_2\left[\frac{U^{2\beta}_{\delta_1\rho_1} - \alpha'_2 U^{1\beta}_{\rho_1\sigma_2}}{U^{0\beta}_{\rho_1\sigma_2}}\right]$$

$$d\Gamma = \frac{1}{2m_A}\left(\Pi_f\frac{d^3p_f}{(2\pi)^3 2E_f}\right)|\,M($$

$$4\,\mathrm{Re}\left\{\frac{2}{1-\Delta\alpha}\chi(s)\left[\hat{g}^e_v\hat{g}^f_v\,(\right.\right.$$

$$p(n) = \frac{1}{\pi\sqrt{2}}\sum_{k=1}^{\infty}\sqrt{k}\,A_k($$

Future

$$\Pi_{j\geq 0}\left(\sum_{k\geq 0} a_{jk}z^k\right) = \sum_{n\geq 0} z^n\left(\sum_{\substack{k_0,k_1,\ldots\geq 0 \\ k_0+k_1+\cdots=n}} a_{0k_0}a_{1k_1}\cdots\right.$$

$$W^{3\beta}_{\delta_1\rho_1\sigma_2} = U^{3\beta}_{\delta_1\rho_1\sigma_2} + \frac{1}{8\pi^2}\int_{a_1}^{\alpha_2} da'_2\left[\frac{U^{2\beta}_{\delta_1\rho_1} - \alpha'_2 U^{1\beta}_{\rho_1\sigma_2}}{U^{0\beta}_{\rho_1\sigma_2}}\right]$$

$$= \frac{1}{2m_A}\left(\prod_f\frac{d^3p_f}{(2\pi)^3}\frac{1}{2E_f}\right)\left|\mathcal{M}\left(m_A - \{p_f\}\right)\right|^2 (2\pi)^4\delta^{(4)}\left(p_A - \Sigma\right.$$

$$4\mathrm{Re}\left\{\frac{2}{1-\Delta a}\chi(s)\left[\hat{g}^e_v\hat{g}^f_v(1+\cos^2\theta) + \hat{g}^e_a\hat{g}^f_a\cos\theta\right]\right\}$$

$$p(n) = \frac{1}{\pi\sqrt{2}}\sum_{k=1}^{\infty}\sqrt{k}A_k(n)\frac{d}{dn}\frac{\sinh\left\{\frac{\pi}{k}\sqrt{\frac{2}{3}}\sqrt{n-\frac{1}{24}}\right\}}{\sqrt{n-\frac{1}{24}}}$$
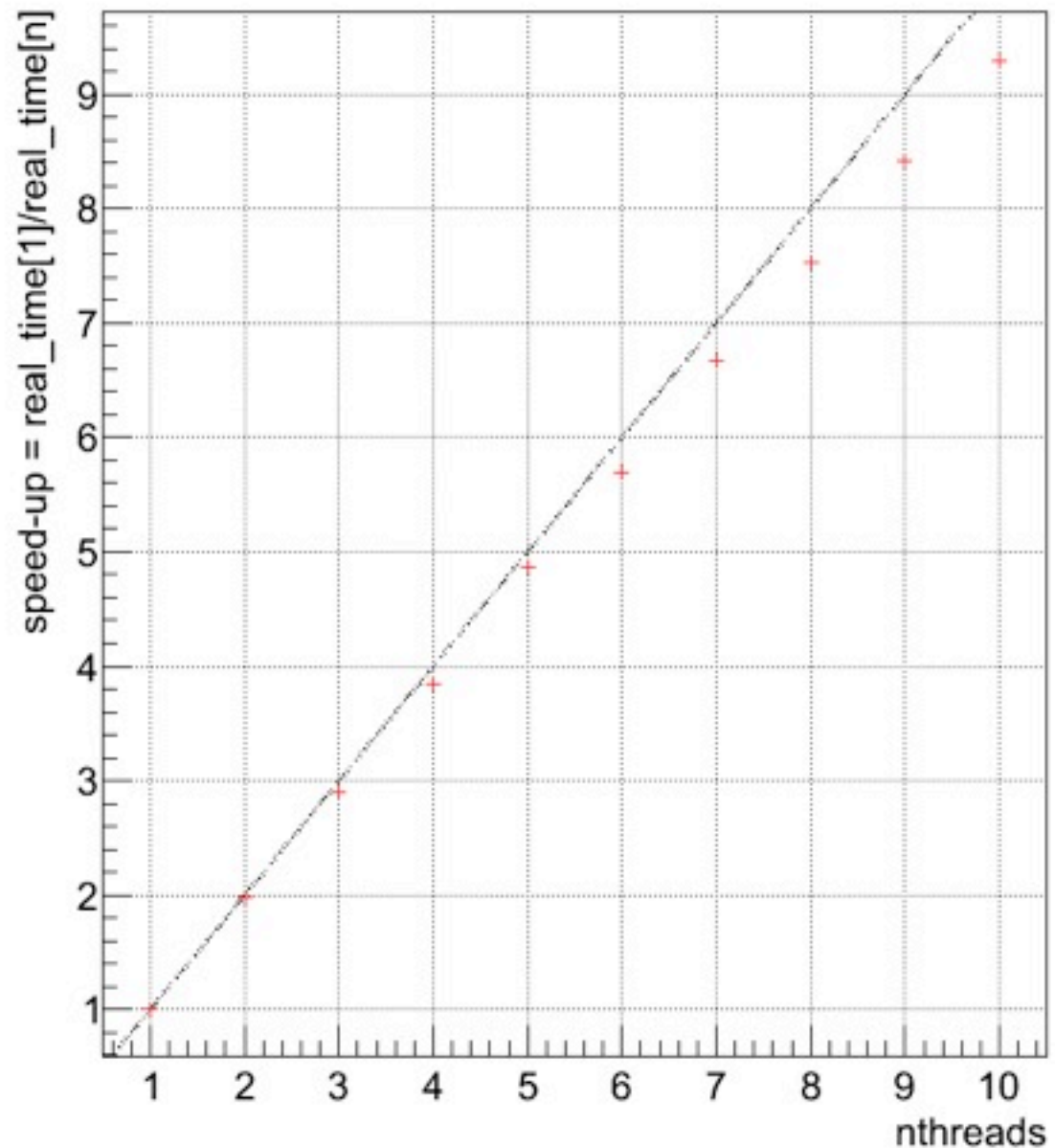
RHIC スピン物理

Parallel Geometry Navigation

- Enabled via: gGeoManager->SetMaxThreads(N)
  - Fast thread ID retrieval implemented via TLS
- Each thread works with its own navigator
  - Navigators holding all state-dependent data
- No locks, no race conditions
  - Stateful data structures migrated to thread data arrays
- Small overhead by replacing direct access
  - Additional calls to TGeoManager::ThreadId()
  - Partially compensated by percolating stateful data in the calling sequence of some internal methods
- Excellent scalability
- See the presentation on Thursday @ 2:20 pm
  - https://indico.cern.ch/contributionDisplay.py?contribId=285&confId=149557

Parallel Geometry Navigation

- Enabled via: gGeoMana...
  - Fast thread ID retrieval
- Each thread works with
  - Navigators holding all
- No locks, no race condit
  - Stateful data structures
- Small overhead by repla
  - Additional calls to TGe
  - Partially compensated
    sequence of some inte
- Excellent scalability
- See the presentation on
  - https://indico.cern.ch/contributionDisp

Speed-up

Eve Event Display

Eve Event Display

Enhancing Concurrency

- Concurrent tasks
    - TTask+TThread and TTaskManager
- Concurrency in I/O
    - Read-ahead thread
    - Parallel merger
- Concurrency and vectorization in math
    - Introduction of the Vc matrix library
- Concurrent geometry navigation
- Thread safe cling
   - Multiple concurrent cling instances

Tablets and Smart Phones

- Currently 320 Million iOS devices, 55 Millions iPads
- All of ROOT ported to iOS (little endian, 32-bit, ARM)
    - To cross compile on OSX to iOS native version: ./configure ios
    - To compile on OSX to iOS simulator version: ./configure iossim
- Android will come later

Implementing ROOT Graphics on iOS

- Ported ROOT's non-GUI graphics on iOS
  - TVirtualPad interface in ROOT::iOS::Pad using Quartz 2D library
- Writing ROOT-based native applications for iOS
  - Demo of ROOT's graphics
  - Browser for ROOT files
- Future developments
  - Components to support users who want to develop for iOS
  - Port ROOT's OpenGL code to GL/ES
  - Use HPC (head coupled perspective) technology for event display (Eve)
    - Uses front facing camera
    - Glasses free 3D display

RootBrowser on the iPad
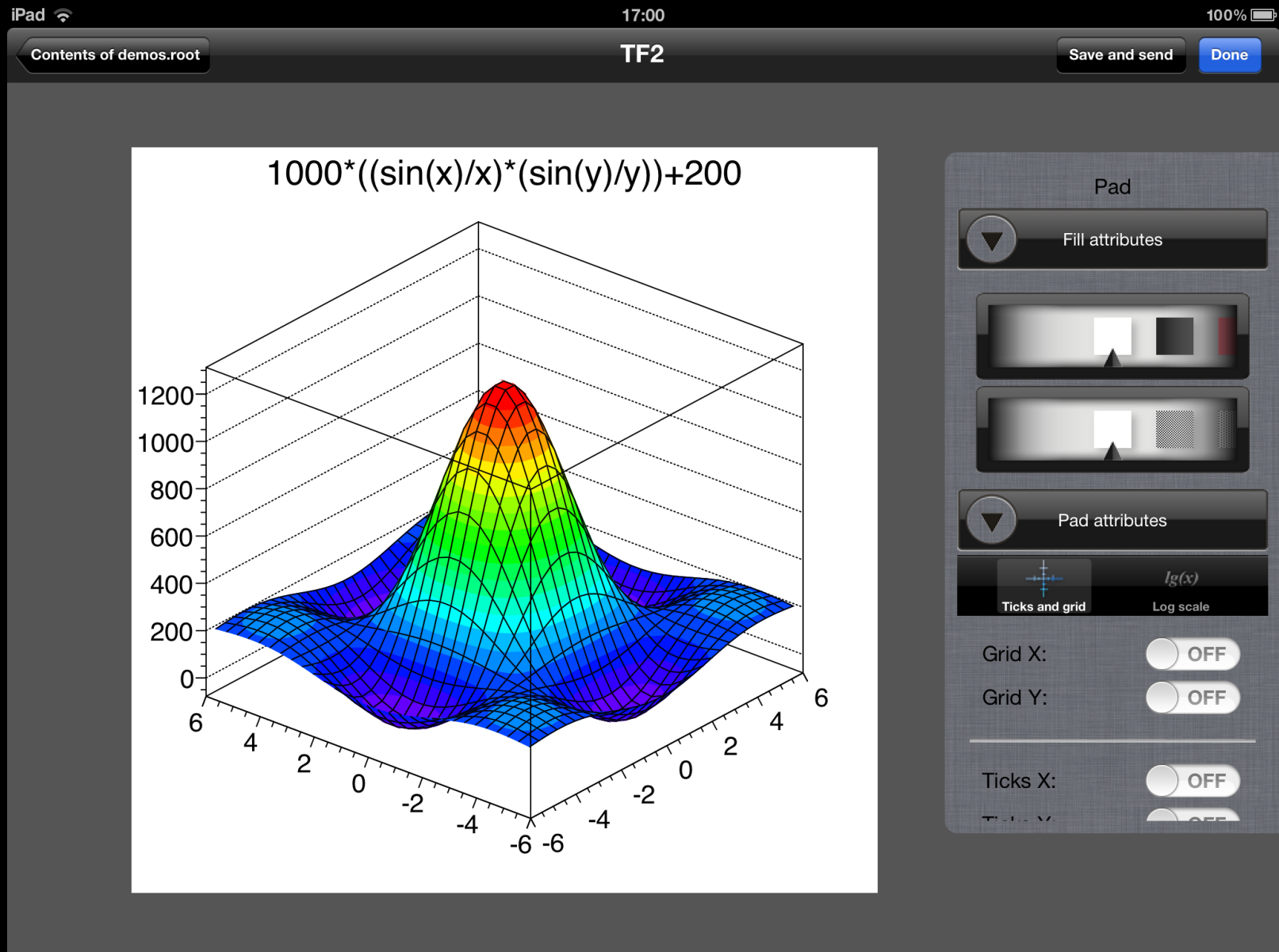
ROOT I/O in JavaScript

- Provide ROOT file access entirely locally in a browser
  - ROOT files are self describing, the "proof of the pudding…"

- See the presentation on Thursday @ 2:20 pm
  - http://indico.cern.ch/contributionDisplay.py?contribId=170&confId=149557
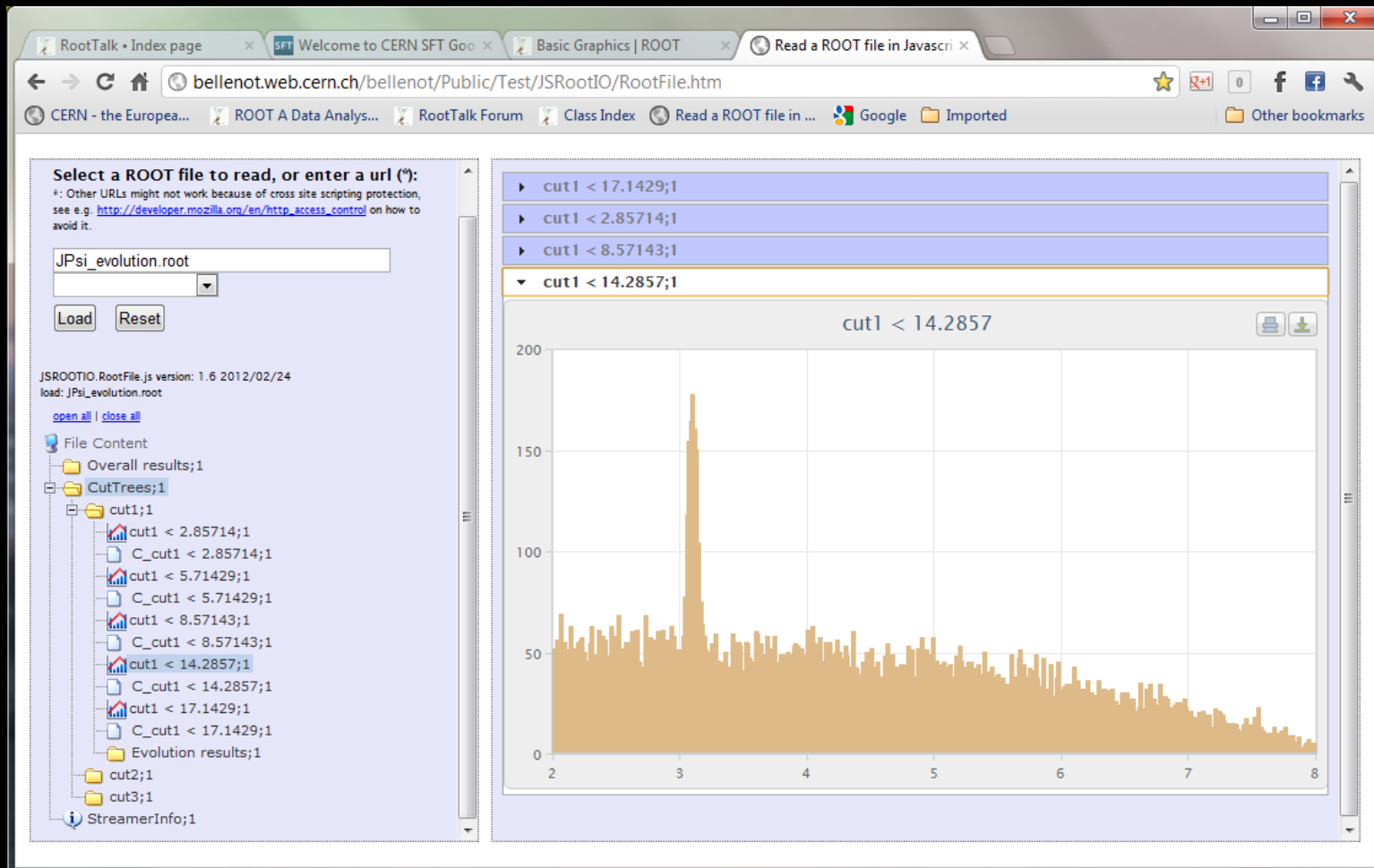
Infrastructure

- Build using ./configure/make and now also cmake
- Static code analyzer
  - Coverity (commercial)
- New continuous build system
  - Electric Commander (commercial)
  - Incremental and full builds and nightly build snapshots for 14 platforms/configurations
- New source code convention checker
  - Eclair (commercial, but uses LLVM inside)
- New documentation system
  - From MS Word to Docbook
- Drupal based website

ROOT Roadmap

- Version v5-34-00 will be released May 30, 2012
  - It will be an LTS (Long Term Support) version
  - New features will be back ported from the trunk
- Version v6-00-00 is scheduled for end of November 2012
  - It will be Cling based
  - It will not contain anymore CINT/Reflex/Cintex
  - GenReflex will come in 6-02
  - It will not have Windows support (expected for 6-02 or 6-04)

Conclusions

- Users and experiments want an absolutely stable ROOT so their carefully tuned analysis are not disturbed when moving to a new version

- At the same time we need to extend to new platforms and incorporate new technologies to make life easier for the users and to keep ROOT on the bleeding edge

- A careful balancing act, but a very exciting one

And don't forget to pass by the CERN PH/SFT booth, located in Kimmel, to see live demos or other amazing developments done in our group