# ROOT Users Workshop 2013 - ”ROOT - The Next Generation”

Sunday 10 March 2013 - Thursday 14 March 2013

Saas Fee

# Book of Abstracts

# Contents

**0**

## Welcome Address by Edmond Offerman

**1**

## Welcome Address by Rainer Flaig, CEO of the Saas Fee Bergbahnen

**2**

## ROOT - State of the Union

**Corresponding Author:** fons.rademakers@cern.ch

**3**

## ROOT Graphics

**Author:** Olivier Couet[1]

[1] *CERN*

**Corresponding Author:** olivier.couet@cern.ch

ROOT 2D and 3D graphics have seen many new developments since the last workshop. We will summarize them, emphasizing the most recent and noticeable ones: ROOT's new default style, new palettes, and its new latex-style text output. We will also give an overview on ongoing and planned projects.

**4**

## Cling - past, present and future

**Author:** Vasil Georgiev Vasilev[1]

[1] *Plovdiv University Paisii Hilendarski (BG)*

**Corresponding Author:** vasil.georgiev.vasilev@cern.ch

ROOT 6 contains a revolution: cling, a new interpreter and reflection database that is built on top of a state-of-the-art compiler infrastructure. We introduce cling, its motivation, requirements and key components.
We explain how cling enables C++11 support and we will discuss ways of making C++ more dynamic and interactive.

We show future perspectives that can enhance the user experience and further shorten the development cycle of user code.

**5**

## ROOT I/O in Javascript

**Author:** Bertrand Bellenot[1]

[1] *CERN*

**Corresponding Author:** bertrand.bellenot@cern.ch

A JavaScript version of the ROOT I/O subsystem is being developed, to browse (inspect) ROOT files in a platform independent way. This allows the content of ROOT files to be displayed in most web browsers, without having to install ROOT, hence giving a direct access to ROOT files from new (e.g. portable) devices in a light way. This presentation will show how to browse ROOT files and display simple graphical objects such as histograms and graphs, using an external JavaScript library with a rendering being being very close to the original ROOT graphics.

**6**

## ROOT 6

**Author:** Axel Naumann[1]

**Co-authors:** Fons Rademakers [1]; Paul Russo [2]; Philippe Canal [3]; Vasil Georgiev Vasilev [1]

[1] *CERN*

[2] *Fermilab*

[3] *Fermi National Accelerator Lab. (US)*

**Corresponding Author:** axel.naumann@cern.ch

As signaled by the first new major ROOT version since almost eight years, ROOT is currently going through a major transition. This presentation will introduce cling, the new interpreter embedded in ROOT 6. It focuses on the consequences of this change, for instance in the user interface, I/O, dictionaries and even math - both in terms of improvements and costs. It explains how this fundamental change enables new, future ROOT features like just-in-time compiled tree formulas or C++11 support.

**7**

## ROOT I/O Review and Future Plans

**Author:** Philippe Canal[1]

[1] *Fermi National Accelerator Lab. (US)*

**Corresponding Author:** philippe.canal@cern.ch

The introduction of cling in ROOT opens several opportunities for I/O including full support for multi-thread I/O and additional performance gains. This presentation will review several of the enhancements introduced in the last few ROOT releases. We will cover the enhancements to TTreeCache that made practical the reading of ROOT file over wide are networks, the introduction of a parallel prefetching mechanism and some of the many performance improvements. It will also focus on some of the exciting features enabled by cling, including a parallel merging facility allowing the fast collation of results produced across many threads, processes or even many computing nodes.

**8**

# PROOF - Lessons learned and future directions

**Author:** Gerardo Ganis[1]

**Co-author:** Dario Berzano [1]

[1] *CERN*

**Corresponding Author:** gerardo.ganis@cern.ch

The use of the PROOF technology for LHC data analysis has been increasing steadily during the last years. PROOF on Demand (PoD) significantly simplified the use of the system on non-dedicated resources, whereas PROOF-Lite provides a quick way to get in touch with the system and an efficient way to exploit the many cores today available on desktops and laptops. Integration with analysis frameworks (e.g. in ALICE) or popular tools (e.g. SFrame in ATLAS) has provided users automatic access to the system.

The first long data taking shutdown period is the opportunity to make the status of the system, present the lessons learned and describe the ongoing and future developments. In this talk we focus on the recent, ongoing and foreseen improvements in the PROOF kernel and PROOF-Lite. These aim mostly at consolidation and performance optimization and include, in particular: an improved packetizing technology, improved merging techniques, a new dataset management model and consolidation of the connection layer.

The status of PoD and its increasing implications in PROOF-enabled (virtual) facilities will be subject of separate presentations.

**9**

# PROOF on Clouds or the Virtual Analysis Facility

**Author:** Dario Berzano[1]

[1] *CERN*

**Corresponding Author:** dario.berzano@cern.ch

PROOF, the Parallel ROOT Facility, is a ROOT-based framework which enables interactive parallelism for event-based tasks on a cluster of computing nodes. A common showstopper for using PROOF has always been its deployment and configuration on a cluster: a big effort has been made to provide generic PROOF analysis facilities available with zero configuration, positively affecting stability and scalability as well. Since a large amount of computing resources are nowadays available through cloud providers in the form of virtual machines, we are presenting the virtual PROOF cluster, a virtual appliance based on CernVM and PoD (PROOF on Demand) ready to be deployed over the cloud either by a single user or by an experiment or organization. We will show the effectiveness of this approach both for sysadmins, which have no or little configuration to do to run the appliance in their clouds, and for end users, which can finally and fully self-support themselves in the event of a code crash.

**10**

## PyROOT: PyCling and Cppyy

**Author:** Wim Lavrijsen[1]

[1] *Lawrence Berkeley National Lab. (US)*

**Corresponding Author:** wim.lavrijsen@cern.ch

With the advent of ROOT 6 and Cling, PyROOT is getting an overhaul of its internals. It will be split in a ROOT-dependent portion (the familiar PyROOT) and two supporting back-ends: PyCling for CPython and Cppyy for PyPy. Through Cling, PyCling gets to be one of the first (if not the first) automatic bindings generator for Python to support C++11. For Cppyy, Cling can deliver the low level details needed for the PyPy tracing JIT to completely remove call overhead, as has been shown to work with Reflex. I will present the status of the transition and the most recent Cppyy benchmark results.

**11**

## ROOT’s graphics: native on Mac OS X.

**Author:** Timur Pocheptsov[1]

[1] *Joint Inst. for Nuclear Research (RU)*

**Corresponding Author:** timur.pocheptsov@cern.ch

Mac OS X traditionally provides a very rich set of APIs and frameworks, particularly in area of 2D/3D graphics. Until recently, ROOT did not have access to these native graphics libraries on Mac OS X.
During 2012 a new graphical back-end was developed for ROOT, as an alternative choice, to the more than 15 year-old X11 based version. It’s a complete implementation of ROOT’s GUI, 2D and 3D graphics, based on Apple’s native APIs/frameworks, written in a mixture of C++ and Apple’s beloved Objective-C.

I will describe the new ROOT modules (macosx, quartz, cocoa), and the motivation and benefits of having a native GUI/graphics implementation, problems and solutions.

**12**

## Redesign of TGeo for concurrent particle transport

**Author:** Andrei Gheata[1]

[1] *CERN*

**Corresponding Author:** andrei.gheata@cern.ch

Concurrency became lately a valuing goal for HEP software for performing better on modern computer architectures. Speeding up the simulation of particle transport is a major R&D objective to cope with the foreseen increase in luminosity for the LHC by 2018. Features like track-level parallelism and vectorization are important levers to move towards high-performance particle transport. The geometry navigation engine behind has to follow in all aspects the upgrades of the transport package in terms of: concurrency, vectorisation and fine grain work provisions to be dispatched to

co-processors like GPGPU or MIC.
The talk will cover the changes already available in TGeo to deal with concurrency, as well as the development plans along this line.

**13**

## Features of the new xrootd client and ROOT I/O plugin.

**Author:** Lukasz Janyst[1]

[1] *CERN*

**Corresponding Author:** lukasz.janyst@cern.ch

Thanks to its scalability and modular structure, the XRootD framework has been successfully integrated into the data storage infrastructure of CERN and many other WLCG sites. Being a key component of EOS and CASTOR systems it provides access to bulk of the LHC data and serves petabytes of storage space for physics data analysis. Extensive usage over several years has revealed many opportunities for improving the scalability and maintainability of the original client software. To address these and better leverage all the features of the xrootd native protocol, we have decided to re-engineer the client.

This contribution presents the new xrootd client, concepts behind its API, its functionality and some performance characteristics. It will also discuss the deployment plan, focusing on integration into the ROOT framework as a new TFile plug-in, and future developments.

**14**

## ROOT and C++11

**Author:** Benjamin Bannier[1]

[1] *Department of Physics and Astronomy, Stony Brook University, SUNY, Stony Brook, New York 11794-3400, USA*

ROOT as a platform is positioned at a demanding intersection of requirements: Data acquisition and analysis in modern HENP asks for often highly optimized implementations, while at the same time many typical users of ROOT are non-experts and are interested in getting correct answers to their scientific questions easily.

The C++11 standard updated the core C++ language with a number of paradigms which allow maintaining a user-friendly, high level of abstraction without overly compromising efficiency, e.g. through shifting work to check type constraints or dependencies from run time to compile time (some of which were already proposed in TR1). New C++11 features like anonymous function objects, automatic type deduction and very easy to use abstractions for multithreaded and asynchronous programming put formerly advanced techniques into the reach of non-experts.

This represents both a challenge and an opportunity for ROOT: to reduce the potential for misuse of its libraries by baffled users, and to build on a common language of modern C++ to communicate design decisions both internally and externally.

Starting from typical use cases problems in representative usage patterns will be shown and possible solutions will be outlined with a focus on usability without compromised efficiency.

**15**

## rootpy: Pythonic ROOT

**Author:** Noel Dawe[1]

[1] *SFU Simon Fraser University (CA)*

**Corresponding Author:** noel.dawe@cern.ch

Python has become the language of choice for high-level applications where fast prototyping and efficient development are important, while glueing together low-level libraries for performance-critical tasks. The PyROOT bindings introduced ROOT into the Python arena, however, interacting with ROOT in Python should not “feel” like you are writing C++. Python also offers a multitude of powerful packages such as SciPy, NumPy, IPython, matplotlib, and PyTables, but a suitable interface between them and ROOT has been lacking. One is left pondering the dilemma of using ROOT or alternatives developed by the Python community. What if it was possible to use the best of both worlds?

The rootpy project is a growing community-driven initiative led by several developers aiming to provide a more pythonic interface with ROOT on top of the existing PyROOT bindings. rootpy takes advantage of Python’s dynamic nature and introspective capabilities to expose more intuitive histogram classes. Plottable objects have properties that alias the ROOT getters and setters, and now optionally accept descriptive strings such as color names. rootpy also provides an interface with matplotlib allowing users to draw ROOT histograms with this popular plotting library if desired. Other major features include the ability to redirect ROOT error messages through Python’s logging system, optionally turning them into Python exceptions. The related root_numpy library can efficiently convert ROOT TTrees into structured NumPy arrays. See rootpy and all related packages at http://github.com/rootpy.

**16**

## Root User Experience in CMS Data Analysis

**Author:** Matthias Ulrich Mozer[1]

[1] *University of Wisconsin (US)*

**Corresponding Author:** mmozer@cern.ch

The presentation will address the experience of non-expert root users in the context of CMS data-analysis. Common workflows will be described and the corresponding challenges discussed. The use of RooFit will take a prominent place due to its importance in several analyses. Suggestions for improvements will be given.

**17**

## Data Analysis in ATLAS from a Developer/User Point of View

**Author:** Attila Krasznahorkay[1]

[1] *New York University (US)*

The analysis model has changed a fair bit in ATLAS since the beginning of data taking in 2009. Our original model, in which analysers would start their analysis from centrally produced Analysis

Object Data (AOD) files, which they could process on the grid using the ATLAS offline software, Athena, and then use the Athena environment until the very last steps of the analysis, was only used in a very few analyses. In our current model a dedicated production team provides simple ROOT ntuple files (called D3PDs) for most physics groups, and most analysers only ever see the ATLAS data in such a format. Also, most analyses are currently done outside of the ATLAS offline software framework, using a simple compilation environment (which we call RootCore) that helps us organise these analysis packages.

In the presentation I will show the upsides and drawbacks of this current model, showing the software techniques that we had to develop to handle LHC data in such simple ROOT files efficiently, and how we organise our analysis code at the moment. This software environment forced us to develop some of the functionalities provided by our offline software in some much simpler format, so that they could be used in a “ROOT only” environment. I will show some examples where additional features from ROOT could help us write analysis code more efficiently/robustly.

There are a number of established features provided by ROOT (for instance PROOF) that only a limited number of ATLAS users take advantage of at the moment. I will try to summarise how these various more advanced ROOT features are being used at the moment in physics analyses.

At the same time, the ATLAS analysis model is under serious discussion at the moment, because the current model is known not to scale to the conditions that we expect for after 2015. One of the most pressing issues we have is to re-think how to organise the ATLAS Event Data Model, preferably taking advantage of the ROOT developments that only appeared in the last couple of years. I will outline the current plans that the experiment has in this area.

**18**

# LHCb Analysis experiences with ROOT

**Author:** Conor Fitzpatrick[1]

[1] *CERN*

The LHCb Experiment relies upon ROOT and ROOT-based applications such as TMVA and ROOFIT for many analysis tasks. This talk will describe how LHCb analysts interact with ROOT in order to obtain results. Physics analysts at LHCb are made up of a broad range of programming experience: Topics presented in this talk will attempt to cover the experiences of seasoned programmers and non-experts alike. Examples taken from recent analyses will be presented, and common issues will be raised.

**19**

# RooFit status and plans

**Author:** Wouter Verkerke[1]

[1] *NIKHEF (NL)*

**Corresponding Author:** verkerke@nikhef.nl

The RooFit toolkit has become a standard tool to formulate probability density functions and likelihood function in HEP. In particular, the development of the ‘workspace’ concept to persist models has greatly advanced the ability to share and combined models, and is the foundation of the RooStats tools and the LHC Higgs discovery analysis. I will review
the current state of RooFit, the workspace and the workspace factory and plans to improve their functionality and performance in the next year, with some particular attention to the new abilities that the cling interpreter in ROOT6 offers.

**20**

## DAVIX: a client library for WebDAV and HTTP-derived protocols oriented to high performance data and metadata access

**Author:** Fabrizio Furano[1]

[1] *CERN*

HTTP-based protocols for data and metadata access are becoming more and more important in the “big data” application domain. HTTP is now the “de facto” standard protocol underlying several big cloud storage systems ( Amazon S3, SWIFT, WebDAV ) and is now commonly accepted in the grid storage world (lcgdm-dav WebDAV, dCache Jetty DAV, WebDAV for Storm).

On top of that, HEP computing (and in general, HPC) needs several features that can technically be achieved with HTTP-based protocols, and that can be at the same time not very easy to design and implement.

Examples of such features include a broad choice of authentication mechanisms (GSI proxies, x509 proxies, passwords, S3 Cloud tokens), support for them in WebDAV, support of the LCGDM extensions for data management, support of session re-use, transparent support of the HTTP redirections in all the primitives, support for vectored IO, metalinks, failover, thread safety and many others.

To the best of our knowledge, finding a third-party client (open source or not) that supports all this is very difficult, as publicly available clients tend to support various permutations of the features, not them all. There are also other examples of client libraries that are good quality implementations of just a low level access layer.
We think that users should be presented a coherent interface that “just works” and that shields them from having to deal with parsing complex XML responses or with security libraries or with too many options for the supposedly simple task of interacting with a data repository.

For these reasons when we needed a complete client to implement parts of the design of our “federated HTTP ecosystem” we decided to use libneon to write a complete and performant client that exposes a simple POSIX-like (open/read/write/close) API that supports all the aforesaid features. We named this client DAVIX.

Now DAVIX is used in the Dynamic Federations project, in GFAL2 and in FTS3. Given its generality, we think that many other applications (e.g ROOT I/O) would get a big benefit using it, exploiting the many data-access related features that it has and its very broad degree of compatibility.

**21**

## The ALICE Data Quality Monitoring Software and ROOT: experiences, limitations and suggested improvements

**Author:** Barthelemy Von Haller[1]

**Co-author:** Adriana Telesca [1]

[1] *CERN*

**Corresponding Author:** adriana.telesca@cern.ch

ALICE (A Large Ion Collider Experiment) is a heavy-ion detector studying the physics of strongly interacting matter and the quark-gluon plasma at the CERN LHC (Large Hadron Collider).
Given its strong requirements in terms of event rate and efficiency, as well as the limited heavy-ion beam time, the online Data Quality Monitoring (DQM) plays an essential role to ensure the good quality of the data.
DQM typically involves the online gathering, the analysis by user-defined algorithms and the visualization of monitored data. AMORE (Automatic MOnitoRing Environment) is the ALICE's DQM framework.
It is written in C++, strongly based on the ROOT framework and interfaced with DATE, the ALICE online framework.
In this presentation, we will show how the integration of ROOT with AMORE has been implemented in order to carry out the data quality monitoring tasks in an efficient manner.
We will show its latest features including the access of ROOT/AMORE objects from the ALICE electronic logbook and the objects visualization through the AMORE Graphical User Interface.
Furthermore, the extensive usage of the framework with thousands of pseudo-ROOT objects produced and published per second will be described as well as the positive and negative outcomes that have arisen from this experience.
Suggestions and limitations associated with the plan of moving to Web-based user interfaces will be presented in the view of the experiment upgrades during the LHC long shutdowns 1 and 2.

**22**

## New developments of ROOT Mathematical libraries

**Author:** Lorenzo Moneta[1]

**Co-author:** Ioan Gabriel Bucur [1]

[1] *CERN*

**Corresponding Author:** lorenzo.moneta@cern.ch

The ROOT Mathematical library have evolved and consolidated in the last few years since the previous workshop. We will present the major new developments such as those in fitting and minimization and the on-going consolidation of the histogram classes. We will show as well the
current work on improving performances of the mathematical libraries using parallelization and vectorization techniques. We will present then the future devolpments with emphasis to those related to the introduction in ROOT 6 of cling

**23**

## RooStats

**Author:** Lorenzo Moneta[1]

[1] *CERN*

**Corresponding Author:** lorenzo.moneta@cern.ch

RooStats is a project providing advanced statistical tools required for the analysis of
LHC data, with emphasis on discoveries, confidence intervals on measurement or exclusions, and combined measurements.
After a brief review of the different statistical tools and their implementation in RooStats,
we will show the experience gained of using them in producing results for physics analysis with real data, such as the Higgs combination results of ATLAS and CMS. We will review new developments which have been included recently in RooStats and we will discuss future improvements, in particular those related to performance optimizations, required to cope with the growing complex models used by the LHC experiments.

**24**

## Git Workflow Best practices

**Author:** Anar Manafov[1]

[1] *GSI - Helmholtzzentrum fur Schwerionenforschung GmbH (DE)*

**Corresponding Author:** a.manafov@gsi.de

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

This presentation is a step-by-step tutorial, that will allow you to test-drive Git with ROOT's sources, showing how to develop code, track changes, and prepare patches. Once you try you will see how your daily ROOT development with a Git repo becomes easier, faster, and safer.

**25**

## Use of the ROOT framework in the LHCb Online

**Author:** Markus Frank[1]

[1] *CERN*

The online system in the LHCb experiment uses ROOT in various areas. ROOT is used in all processes participating in event data processing. The degree of usage varies quite significantly - from the very rudimentary usage of the ROOT plugin mechanism to fully equipped applications filling histograms with data describing online the detector status for monitoring purposes and the display of these data. An increasing number of processes uses the python binding offered by PyROOT to configure these processes. PyROOT also allows to efficiently and quickly manipulate certain corners of the experiment controls system where necessary.
Beside these areas, where the LHCb online team advocated the usage of ROOT, in other areas other technologies were chosen. These deliberate choices like e.g. in the area of persistency of event data from particle collisions will be discussed.

**26**

## Running ROOT as x32-binary

**Author:** Nathalie Rauschmayr[1]

[1] *CERN*

**Corresponding Author:** nathalie.rauschmayr@cern.ch

x32-ABI is an application binary interface, which has been introduced in Linux kernel 3.4. This interface bases on the x86-64 instruction set but uses 32-bit as size for pointers and C-datatype long instead of 64-bit. Thus software can profit from lower memory footprint but also form faster system calls. Several Root-benchmarks have been evaluated in this context and results regarding memory consumption and CPU-time will be shown.

**27**

## ALICE analysis experience with ROOT

**Authors:** Andrei Gheata[1]; Jan Fiete Grosse-Oetringhaus[1]

[1] *CERN*

**Corresponding Author:** andrei.gheata@cern.ch

ALICE is the first experiment at CERN that adopted officially ROOT as baseline from the early beginnings. The whole AliRoot framework is built on top of ROOT classes and profits from its features like: trees, containers, geometry, I/O and histogramming packages. While the skeleton for ALICE data analysis has been formalized by a dedicated framework on top of the TSelector functionality, the users are otherwise free to choose the data structures or ROOT features they consider appropriate for their own analysis programs. The presentation will summarize their experience and wishes in many different ROOT aspects. We will also describe our expectations from ROOT in view of the future evolution of the ALICE analysis framework, ranging from local to distributed data processing.

**30**

## Use of ROOT in Geant4

**Authors:** Andrea Dotti[1]; Ivana Hrivnacova[2]; Witold Pokorski[3]

[1] *SLAC*

[2] *Universite de Paris-Sud 11 (FR)*

[3] *CERN*

**Corresponding Author:** ivana@ipno.in2p3.fr

While the Geant4 framework itself is independent from any choice of analysis tool, the Geant4 user developing his own application can include ROOT analysis in several ways. The simplest is use of g4tools integrated directly in Geant4 since 9.5 release that hides the chosen persistency technology to the user, then the application can use directly classes from ROOT package and finally users can use AIDA compliant tools. All these options are demonstrated in the Geant4 examples.
In this presentation we will give an overview of the new analysis module used as bridge for histogramming and storing ntuples and the examples concerning the use of analysis tools. We will also present the use of ROOT in Geant4 testing, including testing on GRID, and finally the experience with ROOT in multi-threaded programs. The issues experienced with usage of ROOT will be also discussed.

**Summary**:

An overview of the new analysis module in Geant4, the examples concerning the use of analysis tools, the use of ROOT in Geant4 testing and the experience with ROOT in multi-threaded programs will be presented.

**31**

## GooFit: A GPU interface for MINUIT

**Author:** Rolf Andreassen[1]

**Co-authors:** Brian Meadows [1]; Karen Tomko [2]; Michael Sokoloff [3]; Weeraddana De Silva [1]

[1] *University of Cincinnati (US)*

[2] *Ohio Supercomputer Center*

[3] *University of Cincinnati*

**Corresponding Author:** andrear@ucmail.uc.edu

The GooFit framework is a tool for parallelising the evaluation of arbitrary PDFs. By exploiting the massive parallelism of nVidia GPUs, speedup factors of 2-300 can be achieved for real-world physics problems.

**32**

## ROOT in Finance: A Hedge Fund Perspective

**Author:** Jason Ward[None]

**Corresponding Author:** jjasonward@me.com

Systematic trading in the financial markets is a challenging problem. In order to have reliable trading strategies one must acquire and prepare historical financial data, visualise data to understand how things work, develop ideas and models to backtest, and then integrate the framework into an environment that uses the trade recommendations.

We look at an example of how ROOT has been used in finance, from the perspective of having worked a decade in HEP followed by a decade in a hedge fund. Some competing software technologies and frameworks for developing and testing systematic trading strategies will be mentioned, and some thoughts and feedback provided on areas where ROOT can be developed in order to continue to be a viable and attractive solution to this kind of application.

**35**

## Program for statistical comparison of histograms

**Author:** Serguei Bityukov[1]

**Co-authors:** Lorenzo Moneta [2]; Natalia Tsirova [3]

[1] *Institute for High Energy Physics (RU)*

[2] *CERN*

[3] *M.V. Lomonosov Moscow State University (RU)*

**Corresponding Author:** serguei.bitioukov@cern.ch

In note [1] is proposed a method for statistical comparison of histograms. The program which allows to estimate “the statistical distance” between two histograms on the base of this method is realized in frame ROOT system. The first experience of using this program in data analysis is presented in this report.

1. S. Bityukov, N. Krasnikov, A. Nikitenko, V. Smirnova “A method for statistical comparison of histograms”, arXiv:1302.2651 [physics.data-an].

**36**

## The FairRoot framework

**Author:** Florian Uhlig[1]

[1] *GSI - Helmholtzzentrum fur Schwerionenforschung GmbH (DE)*

**Corresponding Author:** f.uhlig@gsi.de

FairRoot is a software framework for detector simulation, reconstruction, and data analysis based on ROOT. It provides basic infrastructure services (e.g: I/O, geometry handling, visualization, etc⋯ ) which allow fast prototyping of detector simulation, reconstruction and analysis codes.
Modular reconstruction and/or analysis is easily achieved by the usage of ROOT TTasks.

Started as a framework for one experiment at the FAIR project, it proves to be suitable for more experiments.
Meanwhile it is used by many GSI/FAIR experiments, as well as by the MPD experiment in Dubna and the EIC collaboration at the Brookhaven National Laboratory.

The design of the framework with focus on some new features like the time-based simulation, the usage of graphics cards, or the database connectivity will be introduced.

**37**

## What to expect from TMVA, the Toolkit for Mulitvariate Analysis in ROOT

**Author:** Helge Voss[1]

[1] *Max-Planck-Gesellschaft (DE)*

**Corresponding Author:** helge.voss@cern.ch

TMVA, the Toolkit for Multivariate Analysis, is a machine learning package which is integrated into the ROOT framework for ease of use by the HEP physicist. Employing and comparing different, sophisticated machine learning algorithms in your data selection of regression tasks has never been easier. In this presentation we will give a short introduction to TMVA and show how to get the most out of multivariate analyses via practical examples. Finally we will present both short and long term plans for the development of TMVA.

**38**

## ROOT and the Gaudi Framework

**Author:** Ben Couturier[1]

**Co-authors:** Marco Cattaneo [1]; Marco Clemencic [1]

[1] *CERN*

**Corresponding Author:** ben.couturier@cern.ch

The Gaudi framework, at the core of LHCb applications, relies on many features of ROOT, from the Mathematical libraries, to the tools for reflection and persistency. While Gaudi's architecture is

under review in order to fulfill the LHCb computing requirements after LS1 and upgrade, significant changes are also announced for ROOT 6. This talk will review the usage of ROOT within Gaudi and LHCb applications, in order to present the features needed by LHCb to migrate to the new ROOT.

**39**

## The Belle II Software Framework and ROOT

**Author:** Christian Pulvermacher[1]

[1] *KIT*

**Corresponding Author:** cpulvermacher@ekp.uni-karlsruhe.de

Belle II is a high-luminosity B physics experiment being built at the
SuperKEKB B factory in Japan.
Until the planned beginning of data taking in 2016, we aim to provide
developers with a stable and user-friendly environment for implementing detector simulation, reconstruction and analysis software.
A new software framework, the Belle Analysis Software Framework 2 (basf2) is under continuing development.

To this effect, we use many different ROOT components, including I/O,
PyROOT, TGeo and Eve.
Our experiences during development will be discussed, as well as our hopes for ROOT6.

**40**

## Cint retrospect

**Author:** Masaharu Goto[1]

[1] *Agilent Technologies International Japan*

**Corresponding Author:** masaharu_goto@agilent.com

Legacy C++ interpreter CINT was developped in Hewlett-Packard Japan in 1991 as an internal tool to support C/C++ language in scientific computation. It was integrated into ROOT framework in 1995. Since then, CINT played a key role as dictionary generator and C++ scripting engine in ROOT.

In this presentation, author of the CINT presents retrospective view; background and motivation of development, how it became key component in ROOT framework, its' evolution and roadblock. Overall, the effort was worthwhile and we use ROOT/CINT as part of semiconductor measurement equipment software today.

**41**

## ROAn, a ROOT based Analysis Framework (not only) for DePFET detector data

**Author:** Thomas Lauf[1]

**Co-author:** Robert Andritschke [1]

[1] *Max Planck Institute for extraterrestrial Physics*

**Corresponding Author:** thl@hll.mpg.de

The ROOT based (Offline) Analysis (ROAn) framework was developed to perform data analysis on data from Depleted P-channel Field Effect Transistor (DePFET) detectors, a new type of active pixel sensors developed at the MPI semiconductor laboratory (HLL). It is highly
flexible and extensible, thanks to ROOT's features like run-time type information and reflection.
ROAn provides an analysis program which allows to perform configurable step-by-step analysis on arbitrary data, an associated suite of algorithms (mainly focussed on DePFET data analysis), and a viewer program for displaying and processing online or offline detector data streams.
The analysis program encapsulates the applied algorithms in interchangeable objects called 'steps' which produce analysis results. The dependency between results and thus the order of calculation is resolved automatically by the program. Also, up-to-date checking is implemented.
Changes of input parameters or updated results are detected and necessary recalculations are triggered. This way the user can concentrate on data analysis without worrying about the consistency of the results.
The viewer program offers a configurable GUI and process chain, which
allows the user to adapt the program to different tasks such as offline viewing of file data, online monitoring of running detector systems, or performing online data analysis (histogramming, calibration,…).
Because of its modular design, ROAn can be extended easily, e.g. be adapted to new detector types and analysis processes.

**42**

## The CMS Data Quality Monitoring Software: experience and future improvements.

**Author:** Marco Rovere[1]

[1] *CERN*

**Corresponding Author:** marco.rovere@cern.ch

The Data Quality Monitoring (DQM) Software proved to be a central tool in the CMS experiment. Its flexibility allowed its integration in several environments: Online, for real-time detector monitoring; Offline, for the final, fine-grained Data Certification; Release-Validation, to constantly validate our reconstruction software; in Monte Carlo productions. The central tool to deliver Data Quality information is a web site for browsing data quality histograms (DQMGUI). In this presentation we will illustrate the usage of the DQM Software in the different environments, its integration in the CMS Reconstruction Software Framework (CMSSW) and in all production workflows, focusing in particular on its interplay with the ROOT package. Special attention will also be dedicated to the DQMGUI server design and features, especially the web-based interactive rendering.
We discuss the main technical challenges and our solutions to them, with emphasis on functionality, long-term robustness and performance.
We will also report our experience in operating the DQM systems over the past years, its limitations and our suggestions in the view of the upcoming CMS upgrades during the LHC long shutdown.

**43**

## PoD: Current status and future plans

**Author:** Anar Manafov[1]

[1] *GSI - Helmholtzzentrum fur Schwerionenforschung GmbH (DE)*

**Corresponding Author:** a.manafov@gsi.de

PROOF on Demand (PoD) is a tool-set, which dynamically sets up a PROOF cluster at a user's request on any resource management system (RMS). It provides a plug-in based system, in order to use different job submission front-ends.
PoD is currently shipped with gLite, LSF, PBS (PBSPro/OpenPBS/Torque), Grid Engine (OGE/SGE), Condor, LoadLeveler, PanDA and SSH plug-ins. It makes it possible just within a few seconds to get a private PROOF cluster on any RMS. If there is no RMS, then SSH plug-in can be used, which dynamically turns a bunch of machines to PROOF workers.

In this presentation new developments and use cases will be covered.
We will talk about a new PanDA plug-in, which uses ATLAS PanDA - a distributed software system developed for the ATLAS experiment. A new worker package architecture with binary payload attachments as well as a dynamic server architecture will be also discussed during the presentation. The talk will also cover revised PoD UI and Server interactions.

**44**

## Micro-CernVM: A Lightweight Virtual Linux Environment for ROOT6

**Author:** Jakob Blomer[1]

**Co-authors:** Gerardo Ganis [1]; Ioannis Charalampidis [2]; Predrag Buncic [1]; Rene Meusel [1]

[1] *CERN*

[2] *Aristotle Univ. of Thessaloniki (GR)*

**Corresponding Author:** jakob.blomer@cern.ch

The next generation of the CernVM virtual appliance, called Micro-CernVM, will reduce the virtual image size and the amount of data that needs to be downloaded in order to bootstrap the virtual machine by at least an order of magnitude. The virtual machine image is only 10MB while the rest of the operating system is downloaded on demand and locally cached by CernVM-FS. As such, Micro-CernVM provides a viable alternative to cygwin in order to run ROOT6 on Windows. We will give a live demonstration of an early development version of Micro-CernVM running ROOT6 on Windows.

**45**

## ProofAna: A General Purpose Analysis Framework

**Author:** Bart Butler[1]

[1] *Harvard University (US)*

**Corresponding Author:** bart.butler@cern.ch

ProofAna is a ROOT-based framework designed to be flexible, fast, and simple to use. Developed at SLAC and now used in several university and lab groups within ATLAS, it combines all the standard analysis framework features with a runtime-modifiable, persistable event data model, templated on-demand branch loading, a simple package system for third-party and user code, and seamless switching between local, PROOF-Lite, PROOF, batch, and Grid jobs. A simple job scheduler allows

multiple analyses to be run in parallel independently on the same inputs, and events can be sorted into several different output files based on arbitrary criteria. The analysis programming interface is extremely straightforward and has been used with much success by several graduate and undergraduate students.

**46**

## Building ROOT with CMake

**Author:** Pere Mato Vila[1]

[1] *CERN*

**Corresponding Author:** pere.mato@cern.ch

ROOT can be built using the CMake build system. The produced libraries and executables are fully compatible to the current standard build of ROOT. One of the initial motivations for this development has been to simply the process of building ROOT for the Windows platform. Using CMake avoids installing the full CygWin environment. This is possible because in fact CMake is a generator that is able to produce from high-level descriptions native Makefiles or integrated development workspaces such as Xcode, Eclipse, Visual Studio, that can be used in the compiler environment of the user's choice.
We will present and highlight the main differences of building with CMake with respect the current build system based on the Module.mk. In particular, in terms of its performance, the different approaches and its overall maintenance load.

**47**

## Mathematica with ROOT

**Author:** Sebastian White[1]

[1] *Rockefeller University (US)*

**Corresponding Author:** sebastian.white@cern.ch

"Mathematica with ROOT" was released on the ROOT pages 2 years ago. It imports data into Mathematica by linking to the CERN libraries with MATHLINK.
It is capable of dealing with all of the commonly used data generated by the large
Physics and Astronomy community that has adopted the CERN technologies.
This project coincided with release 8 of Mathematica so we benefitted from the
customizable import/export features new in 8.

**Summary**:

In this talk I will give an update on analysis of several large data sets from the LHC experiments as well as applications in smaller R&D applications. I will give an update
on its application in the latest versions of ROOT and Mathematica (release 9) and illustrate
the potential role of new technologies in Mathematica and Alpha to high energy physics computing.
There have been several discussions at CERN and elsewhere about the potential benefit to our community of further collaboration with Wolfram. I will also touch on these in my talk.

**48**

## ROOT used in Test and Measurement Industry

**Author:** Masaharu Goto[1]

[1] *Agilent Technologies*

Abstract: In Agilent Technologies Tokyo site, we develop test and measurement equipment for semiconductor industry. Dielectric material is a key element to build semiconductor device and high performance measurement equipment. We use ROOT in a physics research of such material. 2D/3D graphics is a powerful tool to visualize simulation result. Fitting function is used to validate charge transportation formula. Thanks to ROOT, we figured out that known approximation does not correctly represent electrical behavior of dielectrics and came up with a new charge transportation theory. Based on the result, we could build high performance measurement instrumentation using low cost material. In our lab, we also use ROOT as high volume data logging and analysis tool. We will present how we use ROOT.

**49**

## XrootD Roadmap To Start The Second Decade

**Author:** Andrew Hanushevsky[1]

[1] *STANFORD LINEAR ACCELERATOR CENTER*

As the XRootD storage system nears it’s first decade, it has not ceased to make continuous improvements in reliability, scalability, security, and overall usability. The effort is managed by a small but very active collaboration. Today we see XRootD and second party implementations of the XRootD protocol deployed at most LHC sites as well as significant penetration in the astro-physics community. These deployments have been supported by responsive, yet tempered, software development to meet their evolving needs. As part of that support, this talk presents the XRootD road map for the near future to set expectations and receive feedback.

**Summary**:

This presentation will provide the XRootD roadmap for the near future as well as some thoughts of where the roadmap may lead.

**50**

## Exploring plots on the web with WebOOT

**Author:** Peter Waller[1]

[1] *University of Liverpool-Unknown-Unknown*

Sometimes you have a lot of plots. And you want to show them to people.

WebOOT is a vision and an experimental python application which allows looking inside ROOT files and objects (e.g, trees and histograms) from a web browser. The main idea is that a plot you want to make is generated on demand from a URL. WebOOT blurs the distinction between the ROOT file and the file system, allowing you to easily compose and compare plots from many different places. With a surprising use of wildcards, many plots can be realised from the effort invested in just one.

**51**

## Intensity Frontier Experiment Analysis experiences with ROOT

**Corresponding Author:** philippe.canal@cern.ch

**Summary**:

The Intensity Frontier experiments use ROOT in various areas and we will describe various use cases focusing in particular on MINERvA. MINERvA is a neutrino scattering experiment which uses the NuMI beamline at Fermilab and seeks to measure low energy neutrino interactions both in support of neutrino oscillation experiments and also to study the strong dynamics of the nucleon and nucleus that affect these interactions. MINERvA uses ROOT in their online system and their physics analysis group has written a new histogram class that extends ROOT's classes with some really nice features for calculating and displaying uncertainties.

**52**

## TGeoCAD: an interface between ROOT and CAD systems

**Author:** Cinzia Luzzi[1]

[1] *Universita di Ferrara (IT)*

For high-energy physics experiments, in addition to the data analysis part, a very high precision in the description of the detector geometry is essential to achieve excellent results.

The physicists team which performs the simulation of the detector needs to strictly collaborate with the engineers team that works on the mechanical design of the detector. Often, this collaboration is made hard by the usage of different and incompatible software.

Therefore, the necessity to improve the level of communication between physicists and engineers led to the implementation of an interface between the ROOT simulation software and the CAD systems.

The TGeoCAD interface is able to convert ROOT files in the STEP (ISO 10303) format which can be imported and used by many CAD systems.

This talk will describe the design and implementation of the TGeoCAD interface, which has been developed to enabling the use of ROOT files in several CAD systems.

**53**

## TBC

**54**

## R and C++: Seamless Integration using Rcpp

**Author:** Romain François[1]

[1] *R Enthusiasts*

The R environment and language for statistical computing (R Core, 2012) has become the lingua franca of statistical research and applied work, both in academia and industry. The language has evolved from its start at Bell Labs in 1980s to its current form as an established Open Source project with a (closed) core group of committers and a wider (open) group of contributors relying on a common infrastructure provide by CRAN. R uses an internal object model which predates C++ fully implemented in C, using functions and macros.

Rcpp provides the missing link on top of a barebones C API. Rcpp is finding wide adoption within the R community with currently around 100 CRAN and 10 BioConductor packages deploying it.

This talk is intended as a walk through the current Rcpp feature set.

**55**

# From Savannah to JIRA

**Author:** Petya Tsvetanova Petrova[1]

[1] *Plovdiv University Paisii Hilendarski (BG)*

Savannah bug-tracking system retires. A new system –JIRA will replace it. I will report the current status of ROOT project migration in JIRA. Also I am going to describe details of how to use JIRA and what are the advantages of it. I will give details of the new functionalities that can facilitate ROOT users in the working process.

**56**

# ROOT experience from a user at ATLAS

**Author:** Ruggero Turra[1]

[1] *Università degli Studi e INFN Milano (IT)*

ROOT is the standard for high-energy experiments, as ATLAS where it is widely used in data analysis. This talk will describe the usage of ROOT and some related toolkit, as proof, from non-expert users, focusing on the most frequent problems experienced by newcomers and the benefit from the python interface. In addition a comparison to other some new tools is briefly discussed.

**57**

# TTree asynchronous pre-fetching and local caching

**Author:** Elvin Alin Sindrilaru[1]

[1] *CERN*

Optimizing ROOT reading performance over the WAN has become a priority given the various use cases in which analysis is performed on data at remote locations. The presentation focuses on two main techniques for achieving a better WAN performance, first by implementing asynchronous pre-fetching and secondly by providing a simple local caching mechanism which has a dramatic impact on certain workloads. The addition of these two new features is done so as not to affect the current implementation and can be easily enabled/disabled by the user. The talk concentrates on the

design and the integration of the asynchronous pre-fetching, emphasizing the synchronization challenges given the multi-threaded environment. Using the asynchronous pre-fetching we can achieve a reduction of about 25% in the transfer time over WAN in comparison to the current synchronous approach in ROOT.

**58**

## Cling from a ”non-ROOT” perspective

**Author:** Thomas Gahr[None]

While the advantages of using interpreted C++, especially using cling, in ROOT are obvious, it also has great potential for applications outside the ROOT community.
My motivation for doing so, as well as the experience of doing that will be outlined in this talk.