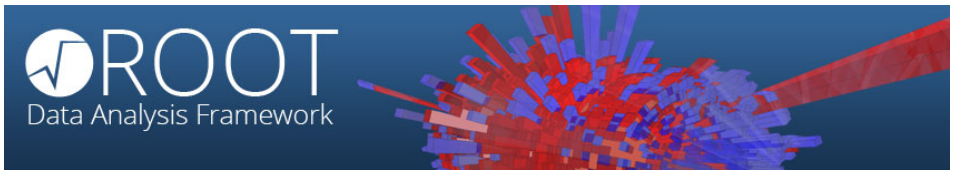
”ROOT Turns 20” Users’ Workshop

Tuesday 15 September 2015 - Friday 18 September 2015

Hotel Schweizerhof, Saas-Fee



Book of Abstracts

Contents

Opening Session / **34**

A few words

Federico Carminati[1]

[1] *CERN*

Corresponding Author(s): federico.carminati@cern.ch

This page intentionally left blank

Presentations / **45**

ALFA: Next generation concurrent framework for ALICE and FAIR experiments

Mohammad Al-Turany[1]

[1] *CERN*

Corresponding Author(s): mohammad.al-turany@cern.ch

The commonalities between the ALICE and FAIR experiments and their computing requirements led to the development of a common software framework in an experiment independent way; ALFA (ALICE-FAIR framework). ALFA is designed for high quality parallel data processing and reconstruction on heterogeneous computing systems. It provides a data transport layer and the capability to coordinate multiple data processing components. ALFA is a flexible, elastic system which balances reliability and ease of development with performance by using a message based multi-processing in addition to multi-threading. The message- based approach allows different parts of the software to run on different hardware platforms (heterogeneous system). The simulation part of ALFA is fully ROOT based, Moreover, ROOT objects are used for persistency and for communication between ROOT based tasks of the software in the reconstruction chain. The status of the development and existing proto-types will be presented in this talk.

Presentations / **24**

An alternative placement method for the geometry package

Angel Perea Martinez[1]

[1] *Consejo Superior de Investigaciones Cientificas (CSIC) (ES)*

Corresponding Author(s): angel.perea.martinez@cern.ch

Solid placement during the construction of simple to moderate geometries implies transformation matrices explicitely coded in the program. An alternative method is
presented, which develops the idea of of virtual connectors present on the surface and
other geometric places of each body. Connectors are then snapped together and/or geometric relationships enforced between pairs/groups of solids, using commands and rules that refer to named points in the solids.

Presentations / **22**

Analyzing LHC experiment software in terms of obsolete memory utilization with a focus on ROOT objects

Author(s): Nathalie Rauschmayr[1]

Co-author(s): Sami Kama [2]

[1] *CERN*

[2] *Southern Methodist University (US)*

Corresponding Author(s): nathalie.rauschmayr@cern.ch

The ROOT framework is used by all LHC experiments in particular for I/O and histogramming. For most of the experiments the memory footprint of their applications represents a major problem forcing them to go from single- to multicore jobs. However, initial benchmark tests have revealed that the main applications of many LHC experiments are able to run with lower memory footprint than what they normally allocate. One typical reason of excessive memory usage is that objects are kept longer in memory than their useful lifetime. In LHC experiment software it has been observed that some large amounts of memory are allocated and used in the initialization phase but not used again or not freed until the end of the process. A new tool (FOM for Find Obsolete Memory) has been developed which helps to spot such unused allocated objects and to detect memory utilization patterns. This talk will explain the tool and show examples for obsolete memory in LHC experiment software with a focus on ROOT objects.

Presentations / **4**

Collaborative development of software and methods for genomic data analysis

Wolfgang Huber[1]

[1] *European Molecular Biology Laboratory (EMBL)*

Bioconductor is an open-source, open-development software project for the analysis and comprehension of high-throughput data in genomics and molecular biology. The project aims to enable interdisciplinary research, collaboration and rapid development of scientific software. Based on the statistical programming language R, Bioconductor comprises over 1000 interoperable packages contributed by a large, diverse community of scientists. Packages cover a range of bioinformatic and statistical applications. They undergo formal initial review and continuous automated testing. I will present an overview of project design and management -both technical and 'social'- and of our efforts to provide a productive environment developers and a positive user experience.
http://bioconductor.org
http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html

Presentations / **39**

DD4hep, a Detector Description Solution for High Energy Physics Experiments

Author(s): Markus Frank[1]

Co-author(s): Andre Sailer [1] ; Frank-Dieter Gaede [2] ; Marko Petric [1] ; Nikiforos Nikiforou [1]

[1] *CERN*

[2] *Deutsches Elektronen-Synchrotron Hamburg and Zeuthen (DE)*

Corresponding Author(s): nikiforos.nikiforou@cern.ch, markus.frank@cern.ch

The detector description is an essential component that is used to analyze data resulting from particle collisions in high energy physics experiments.
We will present a generic detector description toolkit and describe the guiding requirements and the architectural design as well as the main implementation choices.
The toolkit is reusing and combining already existing components and elements from the ROOT geometry package and the Geant4 toolkit to offer a complete and coherent detector description solution aimed to solve all typical problems during the entire experiment life-cycle. All detector description related issues from data processing application required during the experiment life-cyle will be addressed.
The design is strongly driven by easy of use; developers of detector descriptions and applications using them should provide minimal information and minimal specific code to achieve the desired result.
We like to use the opportunity to also report difficulties combining the ROOT geometry package and the Geant4 toolkit to a coherent detector description when different philosophies or contentions appeared.

Presentations / **49**

Event Visualisation Environment of ALICE

Jeremi Niedziela[1]

[1] *Warsaw University of Technology (PL)*

Corresponding Author(s): jeremi.niedziela@cern.ch

Event Visualisation Environment of ALICE (AliEVE) is a tool based on the TEve module of ROOT. It was created over a decade ago and is still actively developed and used in production. Changing experimental conditions, evolving users' expectations and an increasing number of performance and stability issues required a major refactoring of the application during LS1 in view of Run 2. In particular the split of the visualisation and reconstruction code using zeroMQ and TBufferFile and the introduction of the so-called Storage Manager will be presented. In addition, a survey on other technologies and platforms which could be of use for the ALICE data visualisation is being conducted: web technologies, mobile platforms, augmented and virtual reality devices are envisaged to be used both in production and for outreach purposes. Thus we present the possible implications for AliEVE's ROOT-based code.

Presentations / **44**

Evolution of multiprocessing in ROOT

Gerardo Ganis[1]

[1] *CERN*

Corresponding Author(s): gerardo.ganis@cern.ch

Multiprocessing in ROOT has so far meant PROOF, the Parallel ROOT Facility. PROOF was designed to address specifically the case of speeding up processing of ROOT trees, and evolved in facilities such as VAF or PAF to exploit generic resources (clouds included), and PROOF-Lite to exploit the many cores available on a single node.

For broader usage, a more generic approach to parallel execution is envisageable, for example exploiting C+11 features available in ROOT 6. Multi-processing still remains appealing because of the minor requirements it puts on the end-user programming skills.

In this presentation we present the current ideas and recent developments for in-node multiprocessing and the plans for distributed setups, including possible integration with deployment toolkits, such as DDS (http://dds.gsi.de/).

Presentations / 47

Experiences with ROOT for ALICE analyses

Jochen Klein[1]

[1] *CERN*

Corresponding Author(s): jochen.klein@cern.ch

ROOT is used in all stages of data processing of ALICE, starting with mass storage of raw data, through simulation and reconstruction, and organized analyses. The data processing is done both online and offline on the Grid. We will first summarize the usage of ROOT in ALICE, in particular for analyses. We will then discuss our experience while focussing on aspects which could be improved, either in ROOT or the way it is used.

Presentations / 18

Explicitly Data-Parallel Programming with C++

Matthias Kretz[1]

[1] *GSI Helmholtzzentrum für Schwerionenforschung*

Computing hardware is steadily increasing the attainable operations executed per second. However, this increase is only accessible through "proper" parallelization of our software. While the multi-core/multi-thread issue has seen a lot of research and solutions in recent years, the intrinsic parallelism per CPU core (SIMD) has widely been neglected. A look at current hardware shows that our software should take SIMD execution seriously.

Starting from a quick introduction to data-parallelism, I will present how this common software property can be translated to SIMD instructions and thus a considerable speedup of the software. This leads to the underlying programming problem: Our programming languages are still unable to express synchronously data- parallel (e.g. SIMD) execution. I will present the current options of the C++ committee for solving the issue. Vector types, as one of the solutions, allow explicit data-parallel programming via the type system. I will focus on SIMD programming with the Vc library, which is available as part of ROOT. The last part of the talk will cover examples of possible applications and upcoming features for the Vc 1.0 release.

Presentations / 46

FairRoot

Florian Uhlig[1]

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

Corresponding Author(s): f.uhlig@gsi.de

The FairRoot framework is the standard framework for simulation, reconstruction and data analysis developed at GSI for the future experiments at the FAIR facility.
Originally developed only for the GSI experiments it is meanwhile also widely used outside GSI.
The framework delivers base functionality for simulation, i.e.: Infrastructure to easily implement detectors, fields, and event generators.
Moreover, the framework decouples the user code (e.g.: Geometry description, detector response, etc.) completely from the used MC engine, which is achieved using
the TVirtualMC interface of ROOT.
The framework also handles the Input/Output using the ROOT IO functionality
which allows to switch the output on or off in a simple and flexible way.
For reconstruction and/or data analysis the user code is organized in modular tasks
based on TTask. The execution order of these tasks is defined via a so-called steering macro.
This scheme allows a very flexible handling of the reconstruction and data analysis configurations, also
allowing to mix the simulation and data reconstruction stage. The Reconstruction tasks can run separately after simulation or directly on the fly within the simulation.
The modular design of the framework has allowed a smooth transition from the
task based to a message queue based system, which makes it possible to parallelize the execution of the tasks without re-designing or re-writing the existing user code.
Breaking the monolithic design into separate processes only communicating
via messages has many advantages. For example it allows implementing the processes in different programming languages or on different hardware platforms.

The framework with a focus on the basic building blocks and the transition to the message queue based system will be presented.

Presentations / **29**

First experiments with TTree I/O parallelisation

Author(s): Enric Tejedor Saavedra[1]

Co-author(s): Danilo Piparo [1] ; Pere Mato Vila [1] ; Philippe Canal [2]

[1] *CERN*

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

Corresponding Author(s): enric.tejedor.saavedra@cern.ch

The TTree I/O pipeline is one of the parts of ROOT that can potentially benefit from parallelisation. In that sense, this presentation will describe the parallelisation strategy that has been followed to speed up the reading, unzipping and deserialisation of the entries of a TTree. Moreover, the results of some experiments with real trees will be shown.

Opening Session / **35**

Fons and ROOT

Fons Rademakers[1]

[1] *CERN*

Corresponding Author(s): fons.rademakers@cern.ch

Fons and ROOT

Presentations / 8

Go4 Version 5 - a ROOT based online and offline analysis environment

Joern Adamczewski-Musch[1] ; Sergey Linev[2]

[1] *GSI*

[2] *GSI DARMSTADT*

Corresponding Author(s): j.adamczewski@gsi.de

The GSI Object Oriented On-line Off-line system Go4 provides a user environment for online monitoring of DAQ data with ROOT based analysis code. The raw data files can be processed and visualized with the same code also in an offline mode. The interactive Go4 GUI combines ROOT and Qt graphics and can control the parameters of the separate Go4 analysis process by means of a generic inter-task communication layer and a user plug-in architecture. For many years Go4 has been used at GSI and elsewhere for production data analysis of experiments in nuclear and atomic physics, for detector test beam monitoring, and for characterization of frontend-electronics prototypes.

Go4 decvelopment was started in 1999 and has always applied and improved most recent ROOT features, like the TThread classes and the Qt-ROOT graphics interface. Go4 version 5.0 has been released in June 2015. Besides supporting Qt5 and ROOT 6, this major release introduces the ROOT HTTP package for communication between analysis and GUI processes. Moreover, the Go4 analysis process with a ROOT web server offers a dedicated JavaScript ROOT GUI, so Go4 analysis may be controlled by any web browser with GUI elements similar to the established Go4 Qt GUI. Vice-versa, the Go4 Qt GUI may visualize ROOT objects like histograms from any ROOT HTTP server even without Go4.

Presentations / 1

Graphics News: new Palettes, Transparency, Interactive editing, LaTex Dump...

Olivier Couet[1]

[1] *CERN*

Corresponding Author(s): olivier.couet@cern.ch

ROOT graphics had many developments since the last workshop. We will summarize them, emphasizing the most recent and noticeable ones and give an overview on the ongoing and planned projects.

Presentations / 33

Highlights and Analysis of the Answers to the ROOT Users' Survey

John Harvey[1]

[1] *CERN*

Corresponding Author(s): john.harvey@cern.ch

The ROOT team has handed out a questionnaire. We have received about 350 responses - thank you very much for your time!

The questionnaire covered the main areas of ROOT, probing use of different parts and whether new features "make it" into actual usage. It asked for feedback on the frequency of use and quality of support and training. And it asked the obvious question: "Which areas would you like to see improved or enhanced?"

This presentation will analyze and summarize the feedback. It will try to condense the replies to the key messages, what ROOT users appreciate and what they would like to see improved.

The results presented here will serve as a guidance for this week; many of the results will re-surface during this week's discussions.

Presentations / **48**

How to bring Modern Machine Learning to HEP

Corresponding Author(s): tobias.golling@unige.ch

Modern Machine Learning algorithms are currently used in almost all big-data fields: search engines, finance, health diagnostics, image and video recognition, and natural language processing, to name but a few. There is clear evidence that HEP, being a big-data field, will substantially benefit from Modern Machine Learning applications in various areas. TMVA, integrated within ROOT, provides the first point of contact for people in HEP to use machine learning, and it has been used extensively and successfully in HEP. In the last decade, since the launch of TMVA, there have been many essential breakthroughs in the machine learning community, such as Deep Learning. Continued steep improvement in terms of performance, automation, speed, robustness and applicability are expected in the near future (i.e. the LHC lifetime). A proposal will be presented to put HEP in a position to capitalise on the many opportunities offered by Modern Machine Learning.

Presentations / **7**

JavaScript ROOT

Bertrand Bellenot[1] ; Sergey Linev[2]

[1] *CERN*

[2] *GSI DARMSTADT*

Corresponding Author(s): bertrand.bellenot@cern.ch, s.linev@gsi.de

JavaScript ROOT aims to provide interactive ROOT-like graphics in the web browsers. One could read and display data from binary ROOT files. Or data can be obtained from JSON representation, produced with TBufferJSON class. JSROOT also implements user interface for THttpServer class.

Presentation will focus on:
- overview of functionality and supported ROOT classes
- different use-cases
- possibility for user classes support

Many useful links can examples can be found on https://root.cern.ch/js/

Github repository https://github.com/linev/jsroot/

Presentations / 41

Julia: a fast dynamical language for technical computing and data analysis

Joosep Pata[1]

[1] *Eidgenoessische Tech. Hochschule Zuerich (CH)*

Corresponding Author(s): joosep.pata@cern.ch

Technical computing has mostly been dominated by statically-compiled high-level languages such as Fortran, C or C++. These general purpose languages have been time tested and perform well in expert hands. Dynamic languages such as Python interfaced with specialized external C/Fortran libraries are now becoming popular in the scientific community, making it easier to get started with computing even for non-experts. However, in the latter approach, significant effort has to go into developing the "glue", restricting the speed of C and the usefulness of Python. What if there was a language that was fast, powerful and simple? Recently, the Julia language has gained enormous popularity with the promise of allowing for high-level, simple and fast technical computing. I will discuss what makes Julia useful, what Julia can offer to the scientific community and demonstrate how it can be used in data analysis and high-energy physics together with ROOT.

Presentations / 3

Modern C++ Interfaces for ROOT

Axel Naumann[1]

[1] *CERN*

Corresponding Author(s): axel.naumann@cern.ch

ROOT's interfaces are mature - they have served us well for decades! But current C++ is different, and has properties that we need:

- ownership becomes type-safety
- concerns are decomposed, reducing interface clutter
- C++'s standard collections are ubiquitous and performant
- optimizers are at a different level than even 5 years ago (let alone 20)
- current students learn different syntax than students of 10 years ago

While we see the need to react, we do not see a backward compatible path into this future. This presentation will show what ROOT's future interfaces might look like, as the basis for a discussion. It will also present how we plan the migration to these new interfaces. Interfaces that will be covered or at least sketched are

- histograms, showing the benefits of an interface with current C++
- smart pointers resembling ROOT's ownership behavior
- TCanvas interplay as an example for ownership management

The goals are an increase in clarity and simplicity, a reduction of memory issues, and an increase in speed. Come judge!

Presentations / 23

Moving CMS developers and analysis community to ROOT6

David Lange[1]

[1] *Lawrence Livermore Nat. Laboratory (US)*

Corresponding Author(s): david.lange@cern.ch

The CMS experiment relies heavily on the ROOT toolkit for both its core software and analysis functionalities. In preparation for data taking in 2015, CMS has completed its transition to use ROOT v6 in close collaboration with the ROOT development team. In this presentation, we will discuss recent development work and experiences based on our use of ROOT6. These include 1) developments and constraints for using ROOT in the CMS multithreaded framework, 2) optimization of CMS data storage, and 3) analysis, e.g., object identification algorithms, recently developed based on ROOT capabilities.

Presentations / 10

Object oriented data analysis at the BGO–OD experiment

Oliver Freyermuth[1]

[1] *Universitaet Bonn (DE)*

Corresponding Author(s): oliver.freyermuth@cern.ch

The BGO–OD experiment at the ELSA accelerator facility at Bonn is built for the systematic investigation of meson photoproduction in the GeV region. It uniquely combines a central, highly segmented BGO crystal calorimeter covering almost 4π in acceptance and a forward magnetic spectrometer complemented by time of flight walls.

Object orientation is a requirement from the beginning to handle the diverse set of involved detectors. As a consequence, starting from the assembly of the event-based data during acquisition up to the level of physics analysis,
ROOT-based datastructures are in heavy use.
All analysis steps are performed with the framework ExPlORA based on ROOT which is optionally complemented by an event generator, Geant4, Geant-VMC, VGM and Genfit2
for monte carlo studies, geometry description and trackfitting.

ExPlORA follows the principles of a plugin and container based data analysis. It offers both a consistent interface structure for plugin development in C++
and a versatile and performant XML-based configuration language which abstracts all steps from filtering the data up to the visualization with histograms or an event-display.

The very portable analysis software can interface with several SQL databases, is subject to continuous testing and supports the developer with a large set of customized warnings facilitated by the reflection mechanisms offered by ROOT.
Successive analysis and levels of data reduction are facilitated by making use of persistent references and a custom pruning procedure.
The framework is complemented by a set of Qt-ROOT based applications for specialized simulations and data calibrations.

Presentations / **16**

PROOF Analysis Framework

Isidro Gonzalez Caballero[1] ; Javier Delgado Fernandez[None]

[1] *Universidad de Oviedo (ES)*

Corresponding Author(s): isidro.gonzalez.caballero@cern.ch, javier.delgado.fernandez@cern.ch

The PROOF Analysis Framework (PAF) provides end users with a lot a extra features over the pure PROOF environment and brings in a defined workflow to perform a HEP analysis over ROOT trees. It has been designed using a modular architecture so that most of its behavior can be easily and dynamically changed or adapted to particular use cases. It is also fully integrated with ROOT taking advantage of the wide use in HEP research communities and the good performance working with huge data files. Among the added value in PAF there is the automation of the processing chain (configuration, compilation, execution), the performance improvement over a bare implementation (estimated between 3 to 12 times better) and the modularization of the analysis enhancing not only the structure of the code, but also the sharing of key elements of the analysis. Moreover, it integrates other tools to create dynamic PROOF clusters like PROOF Lite, PROOF on Demand, PROOF Cluster or PROOF Cloud. We will show the overall design of PAF, its main functionalities and the performance of the tool.

Presentations / **28**

Packaging ROOT for Fedora and EPEL

Mattias Ellert[1]

[1] *Uppsala University (SE)*

Corresponding Author(s): mattias.ellert@fysast.uu.se

As the maintainer of the ROOT package in the Fedora Linux distribution and in the Extra Packages for Enterprise Linux (EPEL) repository, I have encountered a number of challenges when applying the Fedora packaging guidelines to the ROOT software. In this presentation I will discuss some of these challenges and how they were addressed when creating the packages. I will also bring up some outstanding issues that if solved would make the packaging of the software easier.

Presentations / **31**

Powering a Player-First Culture with Massive Gameplay Data: A Sneak Peek at Data and Electronic Arts

Navid Aghdaie[1]

[1] *Electronic Arts*

As one of the largest game developers and publishers in the world, Electronic Arts serves tens of millions of players on a daily basis. These players interact with us across the world, playing dozens of games covering many genres, running on multiple gaming platforms. At EA, data collected through these diverse interactions is used to drive evidence based decisions across the company and power data services which help optimize the games for delighting the players. In this talk we will look at data usage at EA, the large scale platform that powers it, and closed-loop systems and predictive analytics used for optimizations.

Presentations / 40

Project Everware: running complicated analysis pipelines made easier

Author(s): Tim Head[1]

Co-author(s): Andrey Ustyuzhanin [2] ; Igor Babuschkin [3]

[1] *Ecole Polytechnique Federale de Lausanne (CH)*

[2] *Yandex School of Data Analysis (RU)*

[3] *Technische Universitaet Dortmund (DE)*

Corresponding Author(s): tim.head@cern.ch

Project Everware

This presentation will introduce the Everware project. The primary goal of Everware is to develop a service which allows for single click execution of complicated data analyses in the user's browser.

Users visit an Everware instance, and provide the URL of the repository they wish to execute. Everware then clones this repository, builds a custom environment according to the specifications in the repository, and launches it in a container. The user is given access to the container via their web browser.

Use-cases for Everware include code demonstrations, tutorials, sharing of research code as well as analysis reproduction and preservation. Everware makes ROOT accessible to more people as they can quickly try it out in their browser.

This is a brand new project, with participating in the ROOT Users workshop we want to spread the word about the project and connect with people working on related projects (RootJS, Root as a Service, etc)

Everware makes extensive use of the features of jupyter notebooks which support many different languages as kernels, including cling. Everware is an open source project: http://github.com/everware More details on the ideas and thoughts behind Everware: http://betatim.github.io/posts/project-everware-reusable-science/

Presentations / 9

Python bindings for C++ using PyROOT/cppyy: the experience from PyCool in COOL

Andrea Valassi[1]

[1] *CERN*

Corresponding Author(s): andrea.valassi@cern.ch

The COOL software is used by the ATLAS and LHCb experiments to handle the time variation and versioning of their conditions data, using a variety of different relational database technologies. While the COOL core libraries are written in C++ and are integrated in the experiment C++ frameworks, a package offering Python bindings of the COOL C++ APIs, PyCool, is also provided and has been an essential component of the ATLAS conditions data management toolkit for over 10 years. Almost since the beginning, the implementation of PyCool has been based on ROOT to generate Python bindings for C++, initially using Reflex and PyROOT in ROOT5 and more recently

using clang and cppyy in ROOT6. This presentation will describe the PyCool experience with using ROOT to generate Python bindings for C++, throughout the many evolutions of the underlying technology.

Opening Session / **5**

ROOT Development Roadmap

Pere Mato Vila[1]

[1] *CERN*

Corresponding Author(s): pere.mato@cern.ch

A summary presentation with the main directions of development for the ROOT software, providing pointers to other presentations at the workshop. Feedback from users will be solicited on a number of items.

Presentations / **14**

ROOT I/O: Status and Perspectives

Danilo Piparo[1] ; Philippe Canal[2]

[1] *CERN*

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

Corresponding Author(s): philippe.canal@cern.ch

With the consolidation of cling and the rise of multi thread programming and vector instruction sets, ROOT in general and the I/O subsystem in particular are presented with new challenges. This presentation will review the latest improvement in the I/O and describe some of the solutions to those challenges.

Presentations / **15**

ROOT and NASA

Kerry Lee[1] ; Ryan Rios[1] ; Stoffle Nicholas[2]

[1] *NASA*

[2] *Lockheed Martin*

Corresponding Author(s): kerry.t.lee@nasa.gov

ROOT as an analysis framework is heavily rooted within the Space Radiation Analysis Group (SRAG) at NASA Johnson Space Center. It is extensively used in physics analyses using theoretical models as well as raw-data processing and data-analysis from detectors on-board the International Space Station and future exploration missions.

SRAG's ROOT frameworks span both the C++ and Python implementations as well as the 5.X and 6.X branches. The presentation will address the ways that ROOT is used in the context of data-analysis; common workflows and challenges will also be described.

Presentations / 43

ROOT in ATLAS TDAQ

Jiri Masik[1]

[1] *University of Manchester (GB)*

Corresponding Author(s): jiri.masik@cern.ch

ROOT is extensively used in many areas of the online software of the ATLAS experiment ranging from monitoring applications of various stages of the data taking, monitoring and analysis of trigger algorithms carrying out event selection to serialization of objects reconstructed in the trigger. An overview of the ROOT usage and user experience in the online operation will be reported.

Presentations / 19

ROOT on C++ Modules

Vasil Georgiev Vasilev[1]

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

Corresponding Author(s): vasil.georgiev.vasilev@cern.ch

The feature "C++ modules" is expected to become part of the C++17 standard. A "C++ modules"-aware build system could reduce build times up to 50%. ROOT can use the feature further - to optimize the execution speed and reduce the memory footprint at runtime.

In this talk, I give a brief introduction of clang's implementation of the C++ modules. I present the experimental results in modularizing ROOT's build system and steps towards using the feature at runtime. I describe some of the encountered challenges during the conducted work.

Presentations / 37

ROOT's New Website

Danilo Piparo[1] ; Nefeli Iliana Kousi[2] ; Nefeli Iliana Kousi[3]

[1] *CERN*

[2] *National and Kapodistrian University of Athens (GR)*

[3] *Academy of Athens (GR)*

Corresponding Author(s): nefeli.iliana.kousi@cern.ch, nefeli.kousi@cern.ch, danilo.piparo@cern.ch

ROOT has a new website. We will present some of its new features, including the new reference guide.

Presentations / 30

ROOTaaS: ROOT as a Service

Danilo Piparo[1] ; Enric Tejedor Saavedra[1]

[1] *CERN*

Corresponding Author(s): danilo.piparo@cern.ch

The RaaS project aims to provide a software and computing infrastructure to allow interactive usage of ROOT in the Cloud, leveraging the tools offered by the Jupiter Project.
We describe the service interface offered to the user, based on the Jupyter notebooks and iPython, as well as the technologies of the service backend, which include Linux containers and distributed file systems like AFS and CernVM-FS.
In addition, we demonstrate how it is possible to integrate the aforementioned software technologies with computing services offered by third parties such as reasearch centres like CERN.
Based on such elements, new ways to perform data analysis and to further mingle the C++ and Python languages will be illustrated.

Opening Session / **36**

Rene and ROOT

Rene Brun[1]

[1] *CERN*

Corresponding Author(s): rene.brun@cern.ch

Rene and ROOT

Presentations / **25**

RooFit status & development

Wouter Verkerke[1]

[1] *Nikhef National institute for subatomic physics (NL)*

Corresponding Author(s): verkerke@nikhef.nl

I will present an overview of recent RooFit developments and highlight how the RooFit data modeling concepts have changed the way collaborative physics analysis are performed in large collaborations, driven by the technique of persisted likelihood models. I will also highlight future development ideas that will address scalability and performance for even more complex data analysis projects than are currently undertaken.

Summary:

Status and development plans for RooFit

Presentations / **26**

Root-Based Analysis in ATLAS

Nils Erik Krumnack[1]

[1] *Iowa State University (US)*

Corresponding Author(s): nils.erik.krumnack@cern.ch

Over the course of run I ATLAS has developed a suite of analysis tools for working outside of the main software framework. These tools have become widely used throughout the collaboration, as they are lightweight, easy to use and easily portable to most machines that run root. During the long shutdown these tools have seen some major rewrites, as well as the addition of new tools and functionalities.

Summary:

An overview of ATLAS's core analysis software packages, and their evolution for Run 2.

Presentations / 11

Simulating Grid Cells using ROOT

Jochen Kerdels[1]

[1] *University of Hagen*

Corresponding Author(s): jochen.kerdels@fernuni-hagen.de

Grid cells are neurons in the entorhinal cortex of rats and other mammals that exhibit a very peculiar behavior: they fire at periodic locations that cover the animal's environment in a regular, hexagonal lattice. Having a firing behavior that is highly correlated with the animal's location grid cells might provide a rare view on the general principles by which neurons in the higher-order parts of the cortex process information. Based on this hypothesis we developed a computational model of grid cells that postulates a dendritic representation of each cell's input space that is learned by a self-organizing process. We implemented this model and performed extensive simulations using the ROOT data analysis framework. In this context, key features that are essential for our work consist in the automatic serialization / deserialization of complex data structures, efficient and fast storage and retrieval of serious amounts of log data, fast and powerful visualization and analysis, as well as the fully featured C++ interpreter.

Presentations / 21

TFormula, random numbers and more news from the Math Department

Lorenzo Moneta[1]

[1] *CERN*

Corresponding Author(s): lorenzo.moneta@cern.ch

An overview of the recent developments in the Mathematical and Statistical libraries of ROOT will be presented.
These new developments include a new library for random number generation, which include a new pseudo-random number generator MIXMAX, based on Kolmogorov K-system which can be mathematically demonstrated to be a random system.
The new capabilities of the TFormula class of ROOT, based on Cling and recently released will be also shown.

Recent improvements of the histogram libraries will be also presented, including new available capabilities in fitting ROOT objects.
Furthermore, we will show also future developments plans in the ROOT statistical libraries and in particular we will show some planned future extensions of the ROOT multi-variate library, TMVA.

Presentations / **2**

TGeo reloaded - beyond the legacy

Andrei Gheata[1]

[1] *CERN*

Corresponding Author(s): andrei.gheata@cern.ch

Being for more than 10 years in production, the ROOT geometry package is used for the description of several geometry setups in HEP and beyond, being also at the core of many applications which require geometry functionality such as particle transport and tracking, event display, detector description or visualization. Recently multithreading capable, the package was used as baseline for developing GeantV and was one of the main references for extracting navigation functionality into common tools such as USolids and VecGeom. While the main goal is to focus the development to a single geometry engine for HEP, TGeo's legacy features have to be preserved and evolved to delegate some functionality to the high performance VecGeom approach. We describe the plan for this migration as well as the mid term expectations for the TGeo package.

Presentations / **6**

THttpServer class in ROOT

Sergey Linev[1]

[1] *GSI DARMSTADT*

Corresponding Author(s): s.linev@gsi.de

THttpServer class in ROOT implements http server for arbitrary ROOT-based application. It is based on Civetweb embeddable http server and provides direct access to all registered for the server objects. THttpServer also supports FastCGI interface and therefore can be integrated with many standard web servers like Apache.

Presentation will focus on 'advanced' features of THttpServer:

- object methods execution
- command interface
- access control
- support of custom user classes

Online documentation and examples can be found on https://root.cern.ch/js/ and http://web-docs.gsi.de/~linev/js/

Presentations / **17**

The Belle II Experiment: ROOT 6 at the High-intensity Frontier

Thomas Hauth[1]

[1] *KIT*

Corresponding Author(s): thomas.hauth@cern.ch

The Belle II detector will be operated at the SuperKEKB electron-positron collider at the KEK Research Laboratory in Tsukuba, Japan. Currently, the detector components are being integrated and the data taking is expected to commence in the year 2018. The Belle II experiment has a diverse physics program, which is taking advantage of the production of B-mesons at the Y4S resonance, to study flavor physics and possible new physics processes at an unprecedented precision.

To gather sufficient data for precision analyses, huge amounts of collision events need to be stored and processed. Furthermore, large Monte Carlo data sets need to be produced and managed. With the current projections, a disk space of 140 PB of data and Monte Carlo will be occupied after 7 years of data taking.

The ROOT framework is involved at every stage of the data processing of the experiment. The ROOT I/O functionalities are used to serialize complex C++ objects, transfer data from the detector and to save it in files on mass storage systems. Furthermore, ROOT's libraries are extensively used in the reconstruction code to provide a convenient way to implement, among others, linear algebra, function minimization and random number generator functionalities. The TMVA package is heavily used in the analysis stage of the Belle II experiment, which employs machine learning techniques to discriminate between background and signal events.

This talk will focus on the transition from the well-established ROOT version 5 to version 6 in the context of the Belle II experiment. While ROOT version 6 includes significant improvements, foremost the LLVM-based Cling C++ interpreter, the transition on a large, existing code base proved challenging. Some of the specific solutions to properly integrate ROOT version 6 chosen by the Belle II software group will be discussed in detail. The talk will conclude with remarks on areas of improvements concerning the ROOT integration into the Belle II framework and suggestions for further improvement of the ROOT framework itself.

Presentations / 12

The Data Intensive ANAlysis (DIANA/HEP) project

Author(s): Peter Elmer[1]

Co-author(s): Brian Paul Bockelman [2] ; Kyle Stuart Cranmer [3] ; Michael Sokoloff [4]

[1] *Princeton University (US)*

[2] *University of Nebraska (US)*

[3] *New York University (US)*

[4] *University of Cincinnati*

Corresponding Author(s): peter.elmer@cern.ch

This presentation will introduce the Data Intensive ANAlysis (DIANA/HEP) project. The primary goal of DIANA/HEP is to develop state-of-the-art tools for experiments which acquire, reduce, and analyze petabytes of data. Improving performance, interoperability, and collaborative tools through modifications and additions to ROOT, its ecosystem and other packages broadly used by the community will allow users to more fully exploit the data being acquired at CERN's Large Hadron Collider (LHC) and other facilities.

As part of the U.S. National Science Foundation (NSF) Software Infrastructure for Sustained Innovation (SI2) program, DIANA is concerned with the overarching goal of transforming innovations in research and education into sustained software resources that are an integral part of the cyberinfrastructure. Up-to-date information on the project can on the project website (http://diana-hep.org).

Presentations / **13**

The future of ROOT with R

Author(s): Omar Andres Zapata Mesa[1]

Co-author(s): Lorenzo Moneta [2] ; Sergei Gleyzer [3]

[1] *Metropolitan Institute of Technology and University of Antioquia (CO)*

[2] *CERN*

[3] *University of Florida (US)*

Corresponding Author(s): andresete.chaos@gmail.com

R is a widely used programming language for statistical computing and it was written in C language, based on S, developed by Bell Laboratories (formerly AT & T, now Lucent Technologies) by John Chambers and colleagues.
It has a powerful set the packages for linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ...
ROOT R is a package that allows you to use all the machinery of R from ROOT in c ++ integrate both technologies allowing a new development model to create new libraries for data analysis.
Many other developments can be based on this work, so that R has a great future in the coming developments in ROOT.

Summary:

- ROOT R
- Using R packages in ROOT
- RMVA (R TMVA)
- Possible future directions

Presentations / **27**

Web- and Grid based ROOT analysis, and national ROOT analysis tutorials in ATLAS

Arturos Sanchez Pineda[1] ; Arturos Sanchez Pineda[2]

[1] *Universita e INFN, Napoli (IT)*

[2] *Universites de Paris VI et VII*

ATLAS has a wide range of software tutorials targeted at different audiences. Besides the ones organised by the Analysis Software Group, there are ones organised locally for various nationalities. The presentation will show experience with such tutorials in Italy, and how they tie in with the tutorials provided by ROOT and ATLAS.

Summary:

An overview of (national) software tutorials in ATLAS, and of using the web and the grid in for analysis.

Opening Session / **32**

Welcome

Closing Session / **38**

Wrap-up Discussion

Corresponding Author(s): pere.mato@cern.ch

Presentations / **0**

Writing good C++14

Bjarne Stroustrup[1]

[1] *Morgan Stanley*

How do we write C++ code that takes advantage of C++14 so that our code is better, rather than just different? How do we do so on a grand scale, rather than just for exceptional programmers? We need guidelines to help us progress from older styles, such as "C with Classes", C, "pure OO", etc. They need to be articulated rules to save us from each having to discover them for ourselves. Ideally, they should be machine-checkable, yet adjustable to serve specific needs.

In this talk, I describe a set of core guidelines developed to help most C++ programmers. This core can be augmented with rules for specific application domains such as embedded systems and systems with stringent security requirements. I do not believe that there could be a single set of rules that could serve everybody well, but there is a core set of useful and important rules that applies to most C++ use. These rules are prescriptive rather than merely restrictive, and about much more than formatting. I describe what the rules currently cover (e.g., interfaces, functions, resource management, pointers, and concurrency) and show a few examples. I describe some tools and a few simple useful classes that can be used to support the guidelines. The core guidelines and a checker tool reference implementation will be open source projects freely available on all major platforms.

Presentations / **42**

XRootD and ROOT Considered

Andrew Bohdan Hanushevsky[1] ; Andrew Hanushevsky[2]

[1] *SLAC National Accelerator Laboratory (US)*

[2] *STANFORD LINEAR ACCELERATOR CENTER*

Corresponding Author(s): abh@stanford.edu, andrew.bohdan.hanushevsky@cern.ch

This talk will take a historical view of how XRootD came about, the changes made to XRootD to improve ROOT performance, and what real-world performance problems remain. This talk will also

look ahead on what could be done to simplify data access and where data transfer may be heading in a few years from now.