LHCb Needs and Priorities

M. Clemencic on behalf of the LHCb Collaboration

CERN - LHCb

June 14, 2013

Outline

Outline

Use of ROOT in LHCb
Software Framework (Gaudi)
User Analysis

Wishes

ROOT 6 from LHCb PoV

Summary

Outline

Gaudi

- Gaudi is a generic event data processing framework
  - used by LHCb, ATLAS and other non-LHC experiments
- Design principles:
  - separation of Algorithms and Data
  - interfaces between User code and framework
  - separation between transient and persistent data
- Gaudi uses ROOT in a few places
  - persistency
  - interactivity (Python)
  - math functions (matrices)
  - plug-in service
 - analysis (histograms/trees)

Gaudi

- Gaudi is a generic event data processing framework
    - used by LHCb, ATLAS and other non-LHC experiments
- Design principles:
    - separation of Algorithms and Data
    - interfaces between User code and framework
    - separation between transient and persistent data
- Gaudi uses ROOT in a few places
    - persistency
    - interactivity (Python)
    - math functions (matrices)
    - plug-in service
    - analysis (histograms/trees)

Gaudi

- Gaudi is a generic event data processing framework
  - used by LHCb, ATLAS and other non-LHC experiments
- Design principles:
  - separation of Algorithms and Data
  - interfaces between User code and framework
  - separation between transient and persistent data
- Gaudi uses ROOT in a few places
  - persistency
  - interactivity (Python)
  - math functions (matrices)
  - plug-in service
  - analysis (histograms/trees)

Gaudi Structure
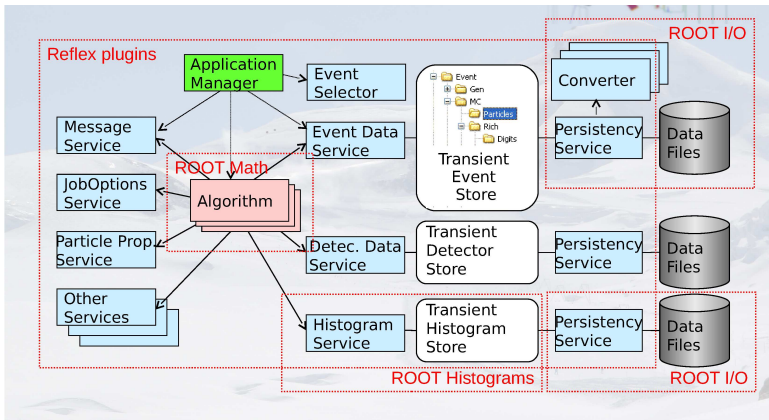
Ben Couturier @ ROOT Workshop

Plug-ins: Reflex::PluginService

- Gaudi relies heavily on plug-ins
  - services, algorithms, tools, converters, . . .
- Custom plug-in service replaced by Reflex in 2006
  - easier maintenance

Plug-ins: Reflex::PluginService

- Gaudi relies heavily on plug-ins
  - services, algorithms, tools, converters, . . .
- Custom plug-in service replaced by Reflex in 2006
  - easier maintenance

Persistency: Reflex dictionaries

- Originally using POOL for I/O
  - dictionaries from Reflex
- Reflex moving into ROOT:
  - simplification
  - transparent for users
  - allowed migration from POOL to ROOT

Persistency: Reflex dictionaries

- Originally using POOL for I/O
  - dictionaries from Reflex
- Reflex moving into ROOT:
  - simplification
  - transparent for users
  - allowed migration from POOL to ROOT

Interactivity: Reflex dictionaries + PyCintex

- Gaudi provides Python bindings
  - easy prototyping and debugging of analysis
- Using Reflex Dynamic Python bindings
  - same dictionaries as for I/O
  - we get both I/O and interactivity in step

Interactivity: Reflex dictionaries + PyCintex

- Gaudi provides Python bindings
  - easy prototyping and debugging of analysis
- Using Reflex Dynamic Python bindings
  - same dictionaries as for I/O
  - we get both I/O and interactivity in step

Math and Analysis

- We use matrices and N-tuples (TTree)
- No revolution expected in this area (correct?)

Math and Analysis

- We use matrices and N-tuples (TTree)
- No revolution expected in this area (correct?)

Platforms

- Required platforms:
    - SLC6 with gcc 4.8 (with C++11 enabled)
    - SLC5 with gcc 4.6 (deprecated)
- Useful platforms (for testing):
    - SLC6 with clang 3.2 (or later) and icc13
      (with C++11 enabled)
- Any other platform are marginal
  (including OSX, Windows, iOS, Android)
- We need compatibility with C++03 and C++11 for SLC5

Platforms

- Required platforms:
    - SLC6 with gcc 4.8 (with C++11 enabled)
    - SLC5 with gcc 4.6 (deprecated)
- Useful platforms (for testing):
    - SLC6 with clang 3.2 (or later) and icc13
      (with C++11 enabled)
- Any other platform are marginal
  (including OSX, Windows, iOS, Android)
- We need compatibility with C++03 and C++11 for SLC5

Platforms

- Required platforms:
  - SLC6 with gcc 4.8 (with C++11 enabled)
  - SLC5 with gcc 4.6 (deprecated)
- Useful platforms (for testing):
  - SLC6 with clang 3.2 (or later) and icc13
    (with C++11 enabled)
- Any other platform are marginal
  (including OSX, Windows, iOS, Android)
- We need compatibility with C++03 and C++11 for SLC5

Outline

Use of ROOT in LHCb
Software Framework (Gaudi)
User Analysis

Wishes

ROOT 6 from LHCb PoV

Summary

Core Functionalities

- We need support for
  - histograms and N-tuples (TTree)
  - TMVA
  - RooFit
- We assume compatibility with xrootd and existing data

Outline

Dear Santa. . .

There are few improvements we would like to have

- better SVG support
- more thread safety
- more parallelization (task-based and supporting TBB)
- better integration of RooFit with ROOT objects
- improved (automatic) I/O optimization

Outline

R.I.P. Reflex

- We know Cint will be replaced by Cling with great benefits. . .

- . . . but we only see Reflex disappearing.

- We do not use Cint or it's dictionaries

- We use PyCintex

R.I.P. Reflex

- We know Cint will be replaced by Cling with great benefits...

- ...but we only see Reflex disappearing.

- We do not use Cint or it's dictionaries

- We use PyCintex

R.I.P. Reflex

- We know Cint will be replaced by Cling with great benefits. . .
- . . . but we only see Reflex disappearing.
- We do not use Cint or it's dictionaries
- We use PyCintex

We Want to Help

- Testing migration to a Reflex-free software
    - replace PyCintex with PyROOT
    - replace the few explicit calls to Reflex with TClass
    - replace the plug-in service (with Axel's help)
- We cannot drop Reflex selection files
    - too many changes
    - no valid replacement in ROOT 5

We Want to Help

- Testing migration to a Reflex-free software
  - replace PyCintex with PyROOT
  - replace the few explicit calls to Reflex with TClass
  - replace the plug-in service (with Axel's help)
- We cannot drop Reflex selection files
  - too many changes
  - no valid replacement in ROOT 5

Outline

Summary

- We are willing to adopt ROOT 6
  - removing Reflex is a pain, but feasible
- Time for validation: it must be ready for Jan 2014
  - we must be able to compile our software and run the tests
- A delay will mean we will stay with ROOT 5 until LS2

Summary

- We are willing to adopt ROOT 6
  - removing Reflex is a pain, but feasible
- Time for validation: it must be ready for Jan 2014
  - we must be able to compile our software and run the tests
- A delay will mean we will stay with ROOT 5 until LS2