ATLAS
LCG Applications Area Feedback

David Quarrie

Introduction

- SEAL

- POOL

- PI

- LCGCMT

- Simulation

- SPI

- Summary

SEAL: Dictionary

- Heavy use of C++ and Python Dictionary

  - Significant contribution to dictionary & Python bindings

- Initially LCGDict for POOL persistency & PyLCGDict for configuration

- More recently PyLCGDict2 for configuration & interactivity

- Haven't yet moved to Reflex or PyReflex

- Good experience with dictionary and code generation tools

  - Now robust and complete

- Don't yet generate all member functions by default

  - Need more studies of space and performance (build time)

- Note that PyROOT developed by Wim now shipped as part of ROOT

SEAL: Plugins

- Migration to use SEAL plugin management underway but slow

  - Used by LCG components themselves (POOL) and within ATLAS Geant4

  - Partially because we have a functional alternative (Gaudi component manager)

  - Partially because a change would not necessarily be backwards compatible

    - Large number of configuration files to change

  - Migration coupled to LCHb plans

- Some problems with cache management & search path

  - E.g. discovered problem between AFS & kit-built installations because of dependency handling differences

  - Building cache files still not fully automated

SEAL: Misc.

- ATLAS doesn't use provided CLHEP dictionary
  - Because we're still using CLHEP 1.8
  - Generates its own (and uses it heavily)
- Concerns about granularity and release couplings
- Some use of Math Library components
  - Particularly Minuit
- We've put on hold a migration to CLHEP 1.9 and 2.0 until the plans for new physics vector and linear algebra libraries are better understood
  - Since migrating to CLHEP 1.9 means we cannot easily evolve existing reconstruction data (easier to recreate) we don't want to move twice
  - Survive for a few months with CLHEP 1.8 (already build our own Geant4 installations)

POOL: Persistency

- Fully committed to POOL

  - Both ROOT and Relational persistency

  - >100TB in ROOT3 format files

  - About to start significant creation of ROOT4 format files

- Limited significant schema evolution experience yet

  - But it will become increasingly important

- Expect to need to write "custom converters" as part of overall strategy

  - Automatic evolution; custom converters; regenerate data

  - Clearly need read capability from raw data for essentially lifetime of experiment

- Generally POOL is robust and complete

  - Small problems still being found & fixed

- Some concerns about rate of new & bug-fix releases

POOL: COOL

- Slower start to COOL than we would have liked

- ATLAS is fully committed to this, and is contributing most of the resources

  - Concern as to viability as common project because of lack of involvement from other experiments

- Lots of work being done in the context of both the online and offline on the COOL pre-releases

- We need a timely first COOL release (due end of March) in order to be ready for early ATLAS detector commissioning

POOL: Scalability

- Scalability is major area still to be addressed

- COOL

  - Interest in FroNtier approach

  - Will this be a common project activity or should ATLAS plan on going it alone?

- General

  - ATLAS fully committed to 3D project as a common effort to achieve scalable distributed infrastructure

PI

- Rather limited use within ATLAS

- Histograms

  - Gaudi/AIDA API inadequate for physics use

  - Profile histograms added recently

  - Histogram manipulation still inadequate

  - TDAQ interested in continued use of HTL behind abstract API

- Still use Gaudi Ntuples by default, but plan on migrating

- Generally not sure what the plans are with respect to App. Area support of AIDA APIs

  - This needs to be clarified soon

LCGCMT

- ATLAS uses LCGCMT for all builds
  - But still needs their own overrides/variants
    - Partially because of use of clhep 1.8
- Distribution kit based on features of LCGCMT
  - We generate our own tarballs of external software

Simulation

- Genser

  - ATLAS uses Genser generators where possible

    - Which ones would be like them to add?

  - About to migrate to Genser 1.0

    - But using same generator versions as currently in use

- Geant4

  - Extensive use in production

    - Geant3 being removed from ATLAS code base

  - Robust and good performance (cpu & memory)

- Validation

  - Feedback, particularly in context of the ATLAS combined testbeam has been crucial

SPI

- Savannah

  - Used by both offline & online/TDAQ

  - Primarily as bug-tracking portal

  - Use dominated by developer responsiveness & management

- Testing tools

- External package installations

  - Most external packages now come from LCG hosted installations

- Config/make support

  - Requests from users for config/make support for AA software

  - Is it still in the plan? Timescale?

Apps Area Reorganization

- ATLAS believes that the new LCG Apps Area organization is capable of functioning well

    - But relies on good integration of teams with different philosophies

    - It requires buy-in from all and design changes from both sides

    - Crucial that needs of experiments continue to be met

        - Remember that we have large user communities

        - We also have production & trigger environments as well as physics analysis users

    - Crucial to clarify plans quickly so that we can understand the implications

        - E.g. CLHEP and maths libraries

        - Long term evolution support

    - This is essentially the last chance before LHC startup

Summary

- ATLAS relies heavily on LCG Apps Area tools

    - And has made significant contributions

- Migration to all SEAL tools incomplete

- Release cascade is worrying

- Some concerns about Apps Area reorganization, but believe it can work

- Need to understand the implications better for our own software base