Summary of persistence discussions with LHCb and LCG/IT POOL team

David Malon
malon@anl.gov
Argonne National Laboratory

Joint ATLAS, LHCb, LCG/IT meeting
1 December 2010

Introduction

- ATLAS has been informally reviewing its I/O and persistence architecture
  - And design and implementation and deployment and performance and …
- Motivated by many factors:
  - Improving performance across a variety of storage and access platforms, data products, and use cases
  - Support for increasingly-many-core architectures
  - Refactoring to make implementations cleaner, more consistent, and more maintainable
  - Feature wish lists that did not make it into production before current LHC running
  - …
  - But also by longer-term planning, particularly in the context of the computing side of ATLAS upgrade planning
- One part of this process is reconsideration of our strategy regarding persistence technologies and interfaces thereto
  - Motivation for today's meeting
- POOL is an essential element of that strategy today

POOL—and Gaudi

- What role does POOL play, what benefits do we derive from it, and at what cost?
- What should our long-term strategy be toward POOL and toward the functionality POOL currently provides?
- For functionality we wish to retain, what should our strategy be?
  - Moderated by the realities of expected support levels
- How should we approach functionality that we would like to have in the long term, but currently lack?
  - And where should it go?
- Important for ATLAS to understand better what LHCb is doing (and planning to do), and persistence-related services in Gaudi
  - ATLAS has not taken significant advantage of the evolution of Gaudi persistence-related services after adoption of POOL, for a variety of reasons
- Is there functionality in POOL (or missing from POOL) that both experiments might care about, that might better be supported in Gaudi?
- Are there current or planned Gaudi persistence capabilities that ATLAS should consider adopting or exploiting?

Time scale?

- IMPORTANT: No precipitous change will be made before the start of 2011 running
  - Expect that we can rely upon the current level of POOL support for 2011 data taking (essentially maintenance)
- Propose not to talk about COOL, or about CORAL very much today
  - Insufficient time, and different stories regarding use across experiments
- Purpose today is to improve the technical foundation for longer-term planning
- Brief talks by
  - Andrea Valassi (LCG/IT), POOL status
  - Peter van Gemmeren , POOL components used by ATLAS)
  - Marcin Nowak (POOL components in support of ATLAS TAGs
  - Markus Frank (LHCb), LHCb POOL use and potential plans

  See http://indico.cern.ch/conferenceDisplay.py?confId=114684

POOL at CHEP 2010

Until recently, three* sets of packages were used by the experiments

Comments by Peter? *Talk by Peter* *Talk by Markus* *Comments by DavidM?* *Talk by Marcin*

POOL RootStorageSvc

Store C++ objects into ROOT files via Reflex dictionary

Event data (ROOT files)

Conditions data (ROOT + COOL)

POOL-ORA (ObjectRelationalAccess)

Store C++ objects into relational tables via Reflex dictionary

Conditions data (Oracle DB)

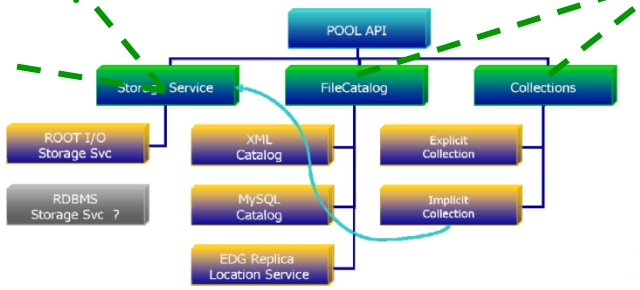
**POOL-ORA was dropped by CMS in Q3 2010*

POOL Collections

Manage collections of POOL C++ objects in different backends

Event collections (ROOT, Oracle DB)

ATLAS 'TAG' database



Evolution in RootStorageSvc usage

(for event data)

	~2003?	~2010?	Future?
DataSvc	  		
PersistencySvc			
RootStorageSvc			 
ROOT			

Observations

- LHCb is considering moving away from POOL after next year's running
- LHCb does not in general see a performance penalty in using POOL
- Interest, rather, is simplification: no need for another indirection layer when the persistence technology will be ROOT in any case
- LHCb has reimplemented POOL catalogs, but no advantage to ATLAS in using this at this point

- ATLAS uses POOL more extensively than any other experiment
- Advantages and disadvantages to moving POOL code into ATLAS code base
- Not much IT effort put into POOL, but do we lose this source of effort entirely if we incorporate the code into ATLAS software?
- No experiment seems to have an architecture that extends much beyond the conversion service layer
  - ATLAS may be the only experiment seriously thinking about this
- No other experiment seems to be thinking much at all about other persistence technologies

Observations

- Impression: if we move away from POOL, it is not clear what persistence services would in fact be shared by the experiments
  - In principle LHCb and ATLAS could share persistence services via Gaudi, but in practice we would likely have separate LHCb and ATLAS ROOT conversion services, for example
    - And there are valid reasons for this … or we would need a serious and concerted effort to do otherwise
- IT is charged with ensuring long-term sustainability of what they support
  - Easiest for them if POOL vanishes—nothing to support
  - Second easiest is if what remains is more fully shared by more than one experiment
  - Third is investing some of their effort in doing what is needed to pass the baton to the experiments
- Technical followups are planned
  - Already a joint ATLAS/LHCB followup on how to use ROOT automatic optimization when the containers that we fill are Tbranches, not Ttrees—what would be involved (in POOL!) and so on
- More work to follow