



# POOL Summary and Plans

---



# Storage Manager Plans



- Mainly performance and consolidation, but...
- Current dictionary loading creates deployment problems
  - All class dictionaries need to be loaded when ROOT file is opened
  - ROOT provides functionality to relax this constraint
    - POOL will work with ROOT team to make lazy dictionary loading available for POOL clients
- Some Object Model support requested
  - Embedded pointer to non-polymorph type - POOL should store objects based on the pointer type
- Recent DataService Review gave some input on enhancements/consolidation requests in that area



# POOL Performance - first cut..



- Has not really been optimised systematically
  - But first results look reasonable
    - We won't be faster than ROOT
    - We won't create smaller files than ROOT
  - But we want to control the overhead we put on top of ROOT - comparing to ROOT in areas where root offers similar functionality
    - POOL collection performance show clearly that POOL insulation overhead can be kept minimal
  - But we provide more functionality and flexibility than ROOT so comparing raw IO speed is sometimes comparing apples with pears



# File Catalog Plans



- Moving into production environment
- Extension to allow for typical the Meta Data evolution use cases
  - Eg A new meta data element is introduced during production
- Composite Catalogs
  - Accessing eg a local writable together with a shared read-only catalog
  - Eg a job reads some user files in addition to any file from the large experiment production



# POOL Collection Futures



- First implementation(s) exist and are being picked up by
- Integration in experiment frameworks just starting
  - How does one find a collection?
  - Is there a Collection Catalog (like the File Catalog)? A central one?
  - Is a File just a special Collection to the end user?
- Role of collections in a grid environment not clear yet - ARDA
- Collection implementation in POOL is a first step
  - But the real issue is not the implementation but rather conceptual
  - Need active experiment involvement in this area



# RDBMS Independence



- POOL should not be directly be coupled to just one RDBMS implementation
- MySQL++ is a pain
  - Need a replacement for many additional reasons
    - Performance constraint on collections implementation
    - Product does not evolve anymore
    - Dependent on internals of the GCC compiler
      - Hard to port mysql++ based code to icc/ecc
- Did a short "market survey"
  - Outcome proposal to move to OTL
  - Started evaluation by providing two prototype implementations of MySQL FileCatalog and Collections
- Prototypes are now part of V1.4 internal releases cycles



# Infrastructure & Testing



- Move to QMtest
  - To align with other LCG projects
- Will soon have to add several new platforms
  - icc ecc and VC for portability check of POOL code and also as additional development platform
- Fully Automate Data Format Regression
  - Highest priority now as experiment data is produced
- Add traceability between bug reports and release contents and release validation tests
  - In collaboration with SPI



# Summary



- POOL has delivered a functional persistency framework and has been integrated into frameworks of CMS, ATLAS and soon LHCb
  - Currently used for test productions in CMS
  - Possibly with more effort than integration teams expected
- POOL as a development team works well and would profit more from insuring stability than additional manpower
  - Some central positions inside POOL are more difficult to back up, but we remained productive even through vacation periods overlapping with experiment integrations
- POOL operates close to its release plan
  - Following "release early, release often" strategy
  - Many experiment requirements have been clarified and agreed only during experiment integration phase rather than upfront
- Many thanks to
  - all developers working on the project for their commitment
  - all experiment integration teams for their patience and very constructive feedback!