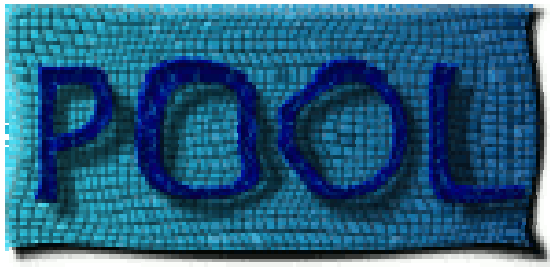




# Persistency Framework - Project Overview

Dirk Duellmann, CERN IT

http://pool.cern.ch and http://lcgapp.cern.ch/project/CondDB/

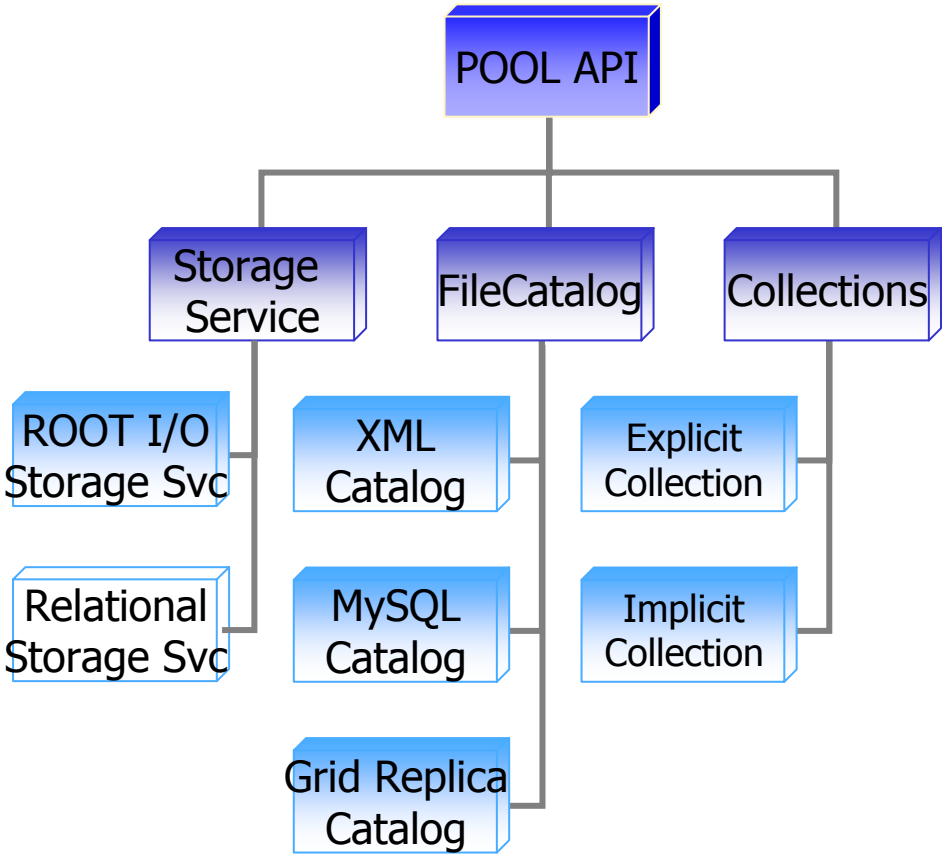
LCG Application Area Internal Review,
March 31, 2005

# The LCG Persistency Framework


- The LCG persistency framework project consists of two parts
  - Common project with CERN IT and strong experiment involvement
- POOL
  - Hybrid object persistency integration object streaming (using ROOT I/O for event data) with Relational Database technology (for meta data and collections)
  - Established baseline for three LHC experiments
  - Successfully integrated into the software frameworks of ATLAS, CMS and LHCb
  - Successfully deployed in three large scale data challenges
- Conditions Database (now called COOL)
  - Conditions DB was moved into the scope of the LCG project
    - To consolidate different independent developments and integrate with other LCG components (SEAL, POOL)
  - Storage of complex objects via POOL into Root I/O and RDBMS backend

# POOL Component Breakdown

- Storage Manager
  - Streams transient C++ objects to/from disk
  - Resolves a logical object reference to a physical object
- File Catalog
  - Maintains consistent lists of accessible files together with their unique identifiers (FileID), which appear in the object representation in the persistent space
  - Resolves a logical file reference (FileID) to a physical file




- Collections
  - Provides the tools to manage potentially large sets of objects stored via POOL
    - **Explicit**: server-side selection of object from queryable collections
    - **Implicit**: defined by physical containment of the objects

# Response to the last review


- Improved Documentation
  - POOL implemented a new documentation scheme based on docbook to create user guide and implementation guides from one source
  - The documentation can still be improved and would profit from involvement of users
  - POOL feature support now close to ROOT
    - Also ROOT 4 was catching up with STL container support
- Schema Evolution
  - Move of POOL 2 to ROOT 4 allowed POOL to profit from the simplified schema evolution support in ROOT
  - First tests in POOL and the experiments show promising results (POOL does not significantly constrain the ROOT functionality)
  - Real confirmation will require experience from experiment deployment of POOL 2

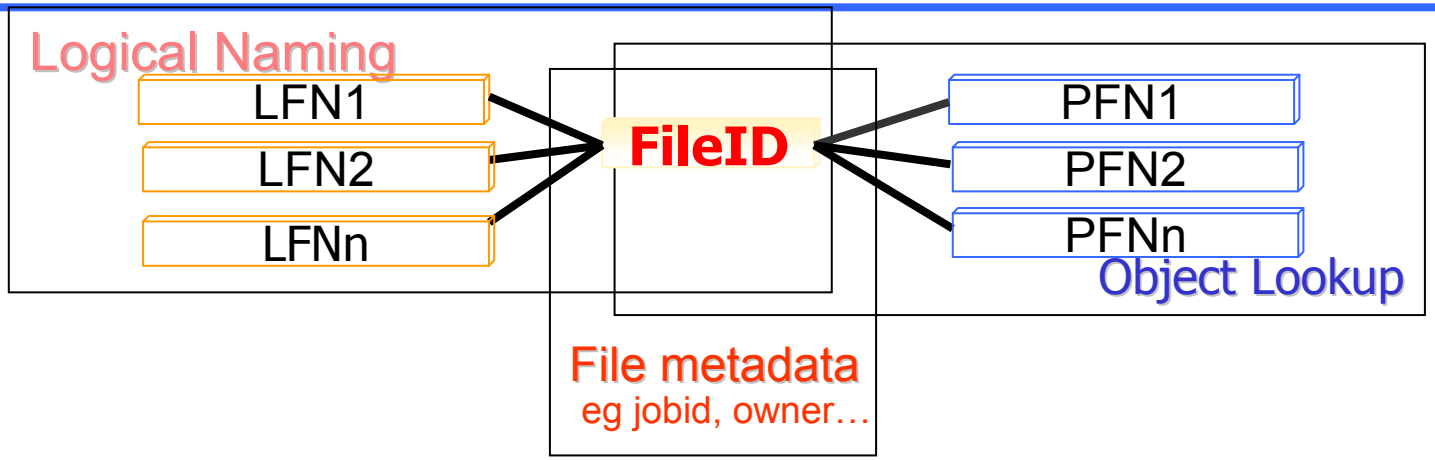
# Responses to last review


- Test coverage
  - POOL has been extending the internal regression testing
    - File format regression (across POOL versions, schema evolution tests, more complex functional tests)
    - SPI tool QMtest has been introduced into POOL
  - Still complexity of experiment test can not be achieved with the available POOL resources
    - Several experiment test have been introduced into POOL, but the dependencies on other experiment s/w was limiting
    - Tight collaboration with integration testing with experiment framework seems more pragmatic and sufficient
- After the POOL internal release testing we typically achieve confirmation on experiment tests of new POOL releases within a few days
  - We believe that this procedure is more economical than spending more effort to relocate all test into POOL
- Optimisation
  - POOL worked with the experiments on optimizations of their POOL <-> framework integration
  - Still a systematic general optimisation of the storage manager component has not been done because of the workload and limited manpower in this area

# Response to the last review


- Files & Collections
  - Main issue here were integration of POOL cross-file references and collections into the analysis environment
  - POOL provided prototype implementations of ref support in ROOT as analysis shell (via a POOL provided plug-in)
  - Neither ARDA (nor LCG AF) turned out to be an forum for collection discussions or integration into analysis frameworks
    - POOL is well connected to the production area but received little input on common model/requirements from the analysis side
    - Result of the maturity/agreement of the computing models in this in this area?
- Requirements are still being actively discussed inside the experiments
  - Analysis with or without the experiment framework and POOL?
  - Support for Refs of non-ROOT destination objects and non-ROOT data (database data) - Required or not?

# POOL File Catalog Model




- POOL adds system generated **FileID** to standard Grid m-n mapping
  - Allows for stable inter-file reference even if lfn and pfn are mutable
  - Several grid file catalogs implementation have since then picked up this model (EDG-RLS, gLite, LFC)
- POOL model includes optional file-level **meta-data** for production catalog administration
  - several grid implementations provide this service (eg EDG-RLS, LFC, gLite)
  - Meant for administration of large file catalogs
    - not for generic physics meta data storage
    - e.g. extract partial catalogs (fragments) based on production parameters
- Catalog Fragments can be shipped (together with referenced files) to other sites / decoupled production nodes
  - POOL command line tools allow cross-catalog +cross-implementations operations
  - Composite catalogs: end-users can connect to several catalogs at once
    - Different implementations can be mixed

# POOL Deployment in the Grid

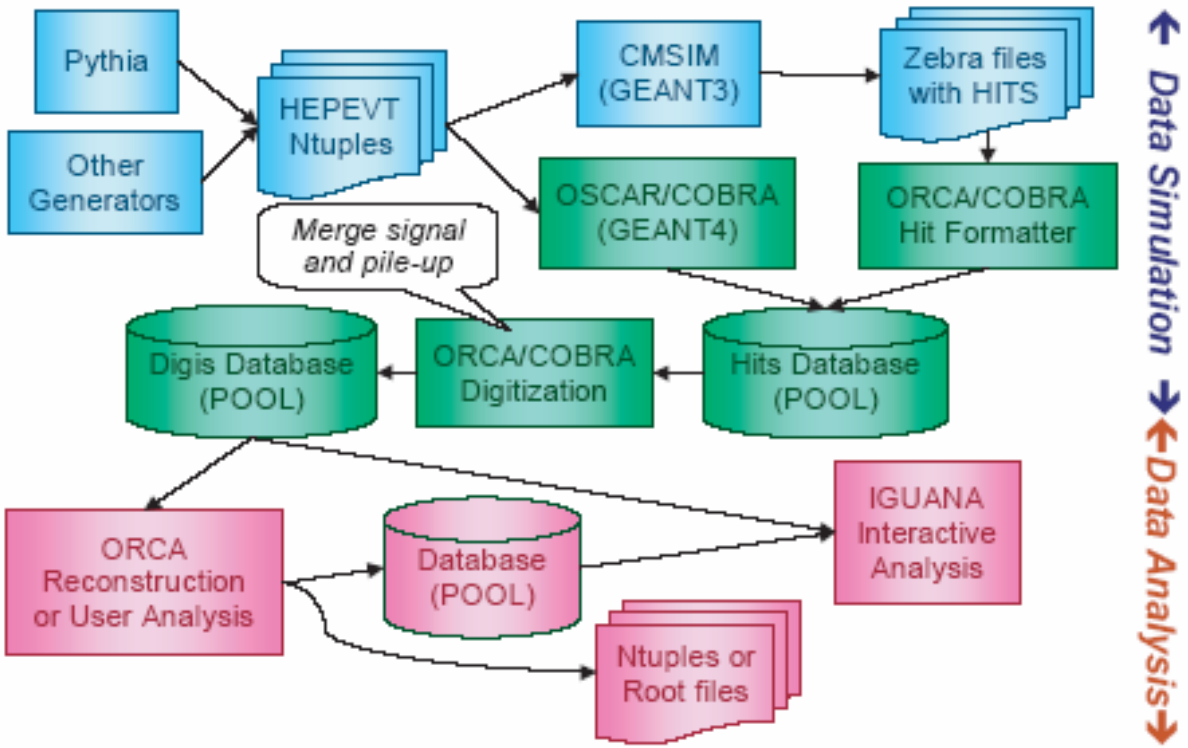

- Coupling to Grid services
  - In 2004 middleware based on the EDG-RLS; Service uses Oracle Application Server + DB
    - Connects POOL to all LCG files
      - Local Replica Catalog (LRC) for GUID <-> PFN mapping for all local files
      - Replica Metadata Catalog (RMC) for file level meta-data and GUID <-> LFN
      - Replica Location Index (RLI) to find files at remote sites (not deployed in LCG)
        - ☹ Resulted in a single centralized catalog at CERN (scalability and availability concerns)
  - Several newer grid catalogs in the queue
    - LFC, gLite, Globus RLS teams will provide implementations of the POOL interface

☝ **But Grid-decoupled modes also required by production use-cases**

- XML Catalog
  - Typically used as local file by a single user/process at a time
    - no need for network
    - supports R/O operations via http; tested up to 50K entries
- Native MySQL Catalog
  - Shared catalog e.g. in a production LAN
    - handles multiple users and jobs (multi-threaded); tested up to 1M entries

# CMS DC04

- Demonstrate the capability of the CMS computing system to cope with a sustained rate of 25Hz for one month
- Started in March 2004 based on the PCP04 pre-production (simulation)
  - Reconstruction phase including POOL output concluded in April 2004
- Distributed end-user analysis based on this data is continuing




| | digitization | reconstruction |
|---|---|---|
| Total amount of data (TB) | 24.5 | 4 |
| Throughput (GB/day) | 530 | 320 |
| Tot num of jobs (1k) | 35 | 25 |
| Jobs/day | 750 | 2200 |

# CMS DC04 Problems

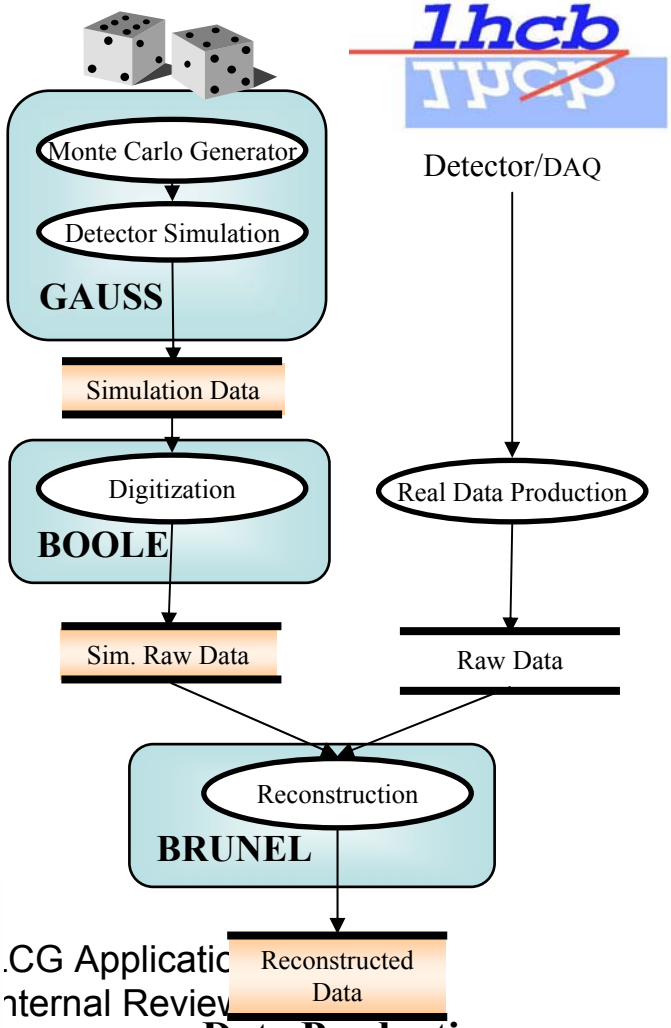
- RLS backend showed significant performance problems in file-level meta-data handling
  - Queries and meta data model became concrete only during the data challenge
    - GUID<->PFN queries 2 orders magnitude faster on POOL MySQL than RLS
    - LRC-RMC cross queries 3 orders magnitude faster on POOL MySQL than RLS
- Main causes:
  - overhead of SOAP-RPC protocol
  - missing support for bulk operations in EDG-RLS catalog implementation
- Transaction support missing
  - Failures during a sequence of inserts/updates require recovery "by hand"
- Basic lookup / insert performance satisfactory
- The POOL model for handling a cascade of file catalogs is still valid
  - Good performance of POOL XML and MySQL backends proves this
  - RLS backend problems being addressed now by IT-Grid Deployment Group
- ☺ Good stability of the RLS service achieved!

# ATLAS Data Challenge 2 scale

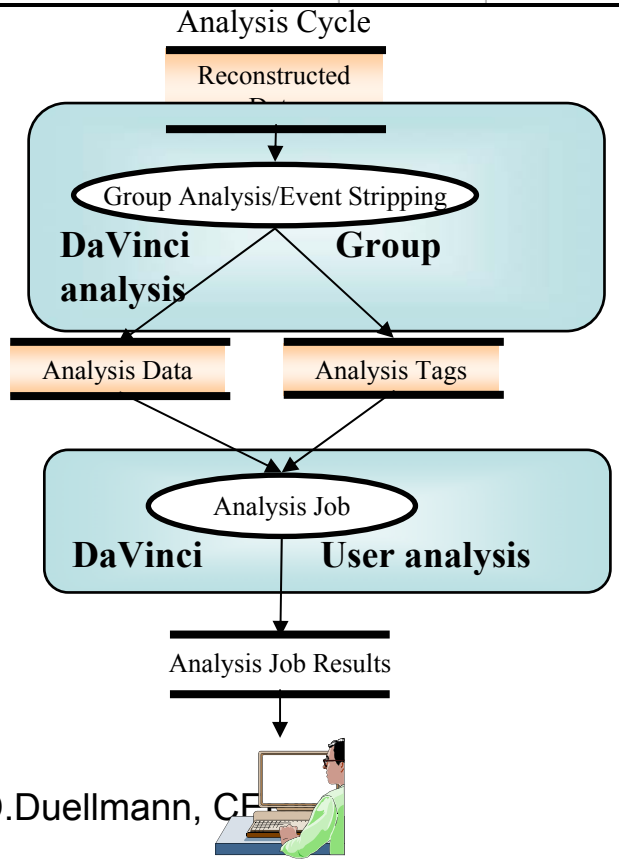
- Phase I: Started beginning of July and still running
- 10^7 events
- Total amount of data produced in POOL: **~30TB**
- Total number of files: **~140K**
- Digitization output is in bytestream format, not POOL
  - This is the format of data as it comes off the ATLAS detector
- Anticipated ESD (October 2004): 700 KB/event→7 terabytes in POOL
  - ESD is currently ~1.5 MB/event, but this will decrease soon
  - 2 copies distributed among Tier 1s implies 14 TB ESD in POOL
- Anticipated AOD (October 2004): 22 KB/event→220 gigabytes in POOL, to be replicated N places (N>6)
- TAG databases: MySQL-hosted POOL collections replicated at many sites
  - "All events" collection ~6 gigabytes; physics collections will be smaller (10-20% of this size)

# Data Processing in LHCb

| File type | # files | # events | Data Volume [TB] | |
|---|---|---|---|---|
| | | | produced | kept in mass storage |
| Simulation data | 791 k | 319 M | 116 | 7 |
| Digitized data | 604 k | 226 M | 128 | 6 |
| Reconstructed data | 348 k | 225 M | 66 | 64 |




Data Production




Data analysis

Gaudi Applications

POOL Files

# POOL Deployment 2004


- Experience gained in Data Challenges is **positive!**
  - No major POOL-related problems
  - Close collaboration between POOL developers and experiments invaluable!
  - EDG-RLS deployment based on Oracle services at CERN
    - Stable throughout the 2004 Data Challenges!
  - File Catalog experience in 2004
    - Important input for the future Grid-aware File Catalogs
- **Successful integration and use in LHC Data Challenges!**
- **Data volume stored: ~400TB!**
  - Similar to that stored in / migrated from Objectivity/DB!