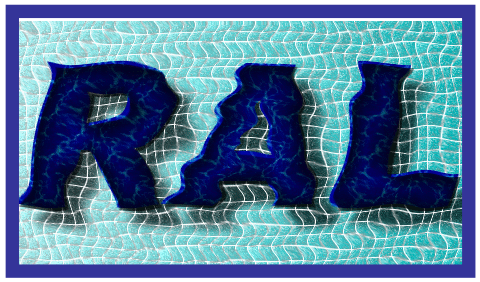
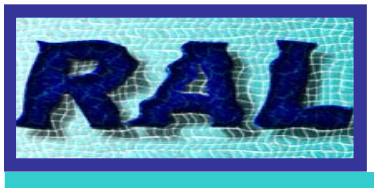
POOL RAL Performance



LCG AA Meting
CERN, 9 March 2005

Radovan Chytracek
CERN/IT/ADC/DP - LCG

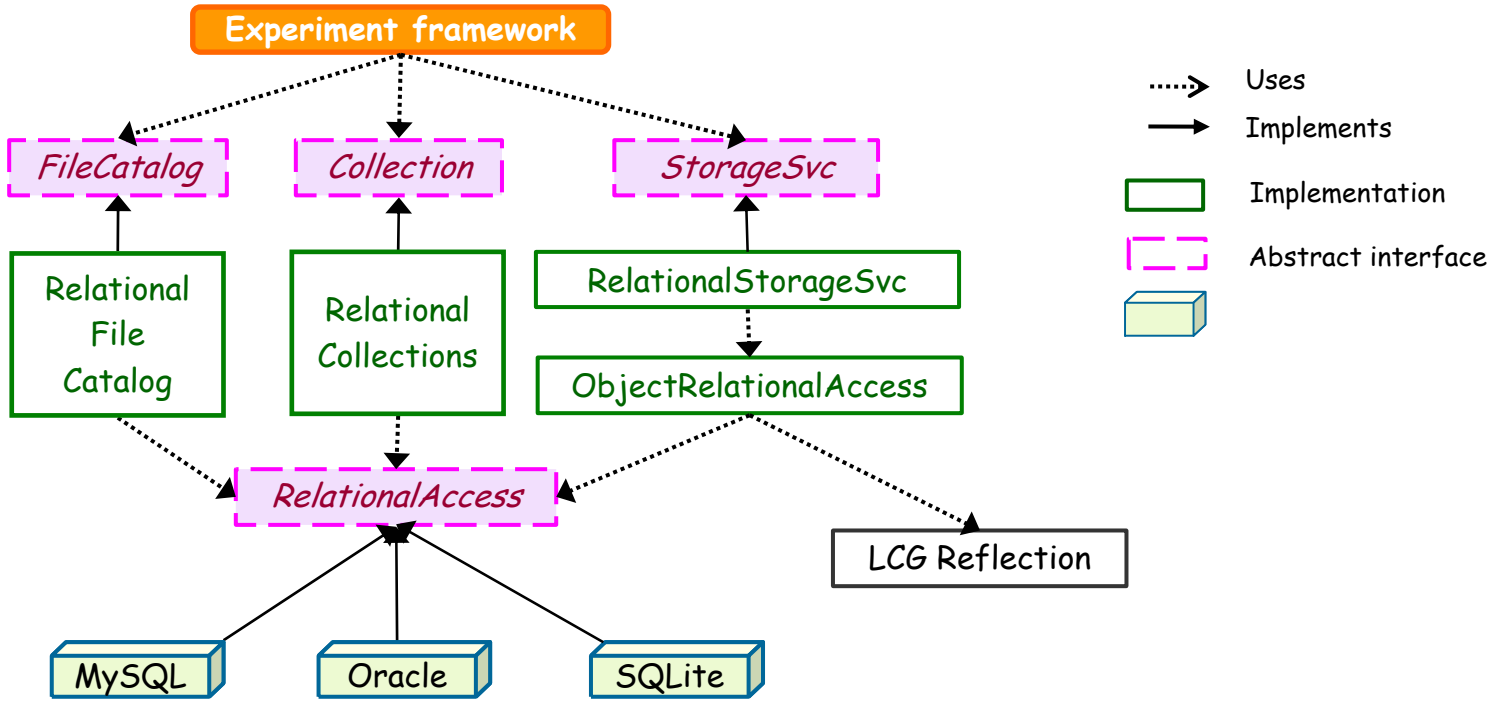


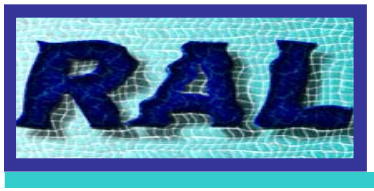
Outline

- Introduction
- Features
- Current status
- RAL vs. native APIs
- Performance Studies
- Relational File Catalog tests
- Recommendations for RAL clients
- New developments
- Summary

Introduction

- RAL is addressing the needs of the existing POOL relational components (FileCatalog, Collection), the POOL object storage mechanism (StorageSvc) and also the ConditionsDB COOL project.





Features

- Abstract, SQL-free API
  - With exceptions of WHERE & SET clauses
- Connection strings storable in a file catalog
  - Example: mysql://raltest/RAL
  - Design decision: no connection credentials in the connection string
  - 3D project is addressing the needs of DB service catalogs
- Schema, table, constraints & index handling
  - Common DDL and meta-data functionality
- Variable binding
  - Named variables syntax supported, e.g. :VARNAME
  - ODBC/MySQL plug-in accepts it but is still positional
    - Translated into ? syntax
- Queries against single or multiple tables
  - Left joins possible
  - Sub-queries (back-end dependent)
- Cursors
  - Forward-only (default)
  - Scrollable
- Bulk inserts
  - Emulated if not supported by the back-end client API or server

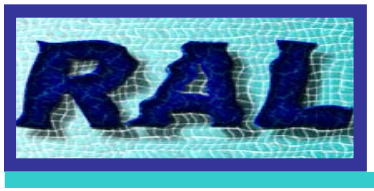
Current Status

- The latest is RAL release is in POOL_2_0_1-alpha
  - Bug fix release
  - Few peformance issues resolved
  - Added support for 64 bit integers

- RelationalFileCatalog
  - Fixes on the variable binding and prefetching

RAL vs. native DB APIs

- RAL is very thin abstraction on top of native DB APIs
  - Zero overhead layer in principle
- Pros:
  - Uniform API for many DB back-ends
  - Usually much less code needed to achieve the same thing than using native API
  - Tries to force the best practices by design
- Cons:
  - Difficult to get into DB back-end specific optimizations
  - Might not fit all use-cases

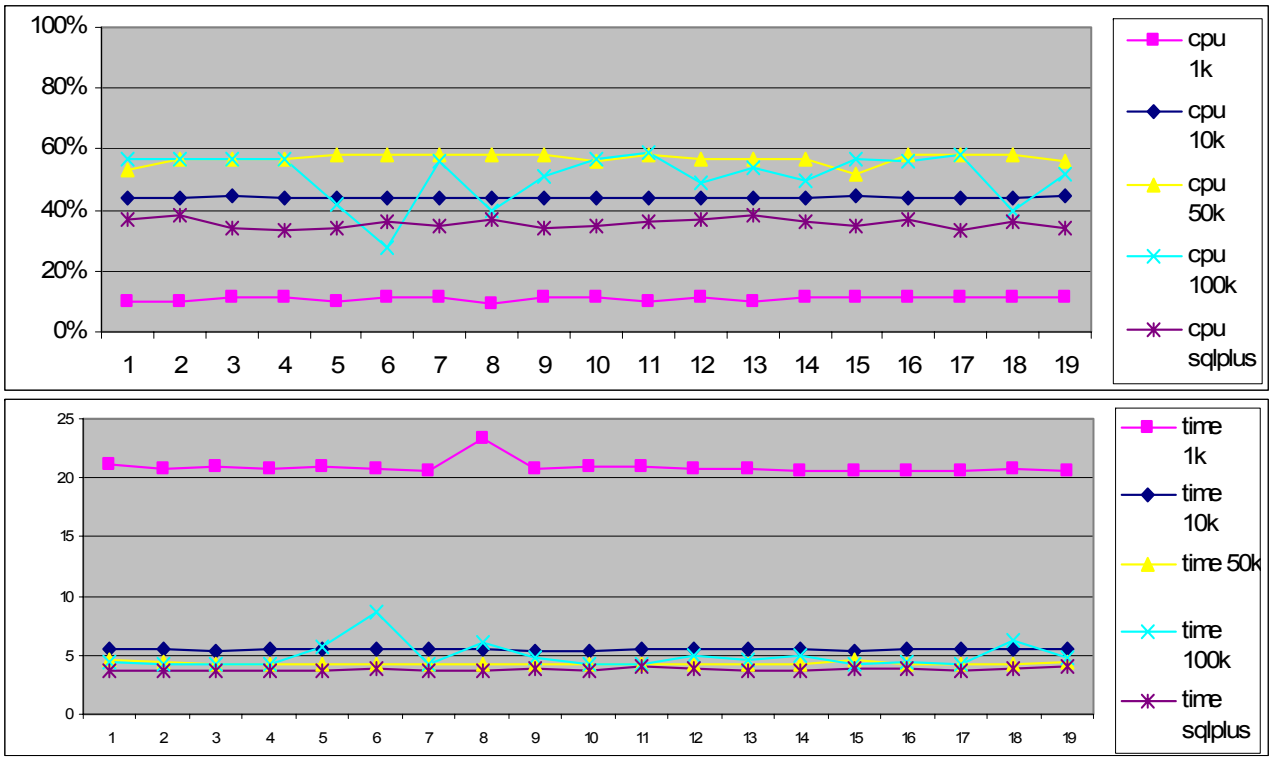


Performance studies

- Conditions database (COOL) tests
  - Performed by team around Andrea Valassi
  - COOL/RAL implementation vs. Lisbon/MySQL
    - COOL/RAL using both of Oracle and ODBC/MySQL
  - Not presented here as results are very preliminary + under very active development
  - Very useful feedback to RAL developers
- POOL Relational File Catalog (RFC) tests
  - Tested against Oracle direct SQL on 10g
    - SQLPLUS
    - Dedicated server node
  - Tested against MySQL direct SQL on 4.0.22
    - Mysql client
    - Desktop box
  - NOTE: The RFC is driven by POOL FileCatalog to issue series of limited queries for large result sets

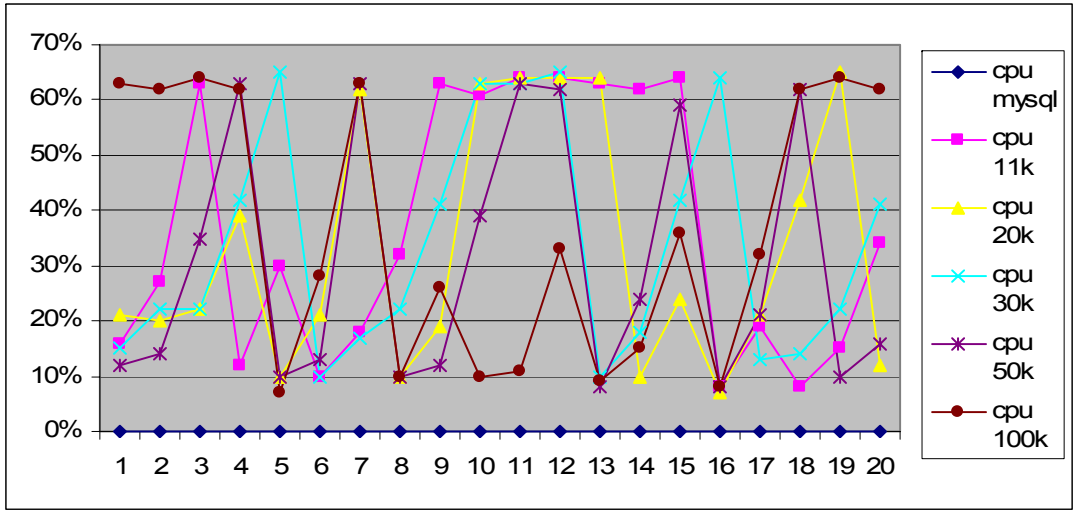
Relational File Catalog case 1 (Oracle)

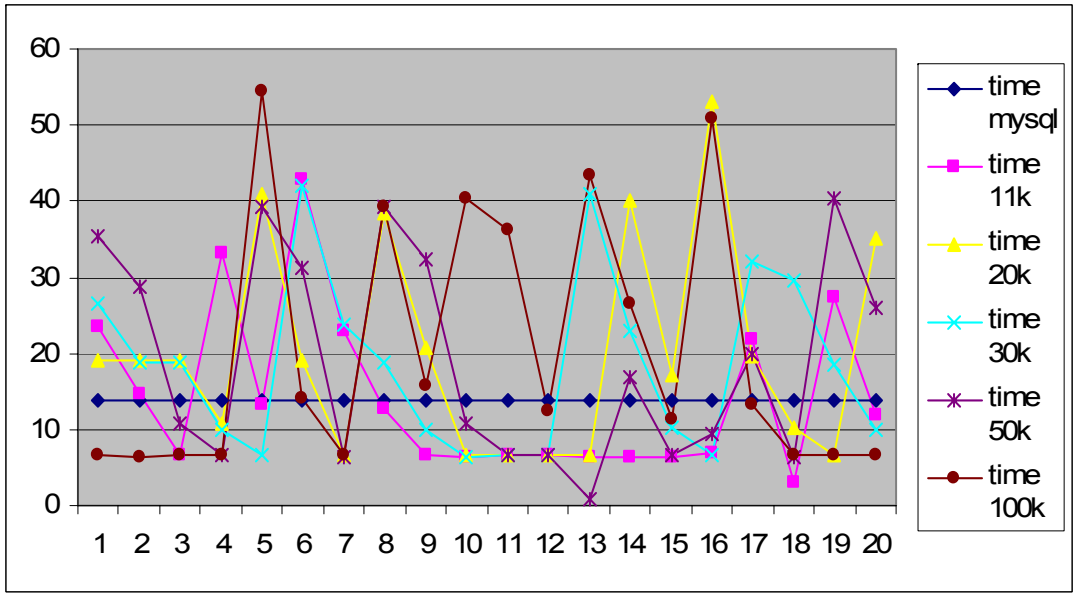
- Query: dataset='eg03_twophoton_born'
- Testing query which usually took down RLS app. server
- Almost 11000 rows in a result set
- A typical metadata query
- 13% slower than SQLPLUS in the best case (50k prefetch buffer)



Relational File Catalog case 1 (ODBC/MySQL)

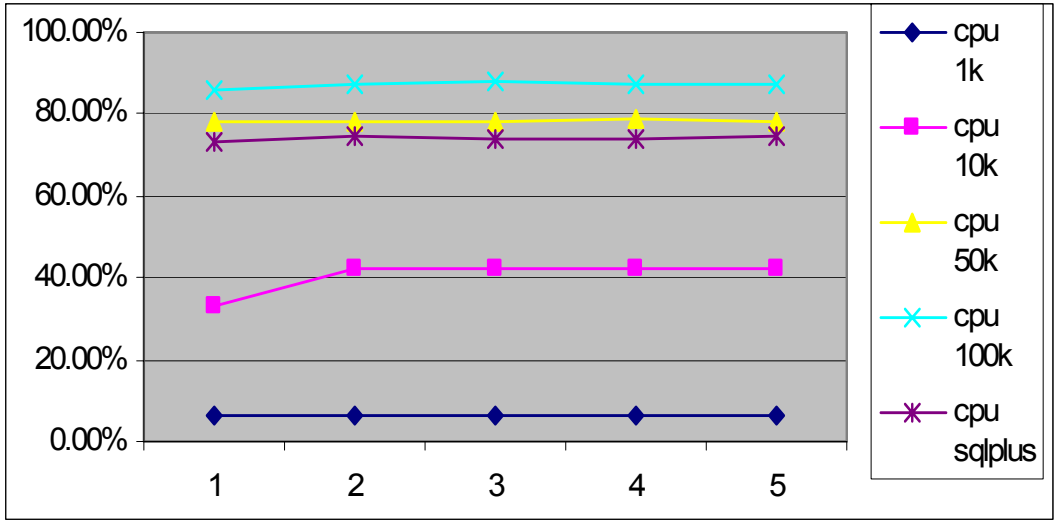
- Query: dataset='eg03_twophoton_born'
- 3-4% slower than mysql client in the best case (11k) but more realistic is 30-40% (20k-30k-50k)
- Not sure one can say here "prefetch buffer"

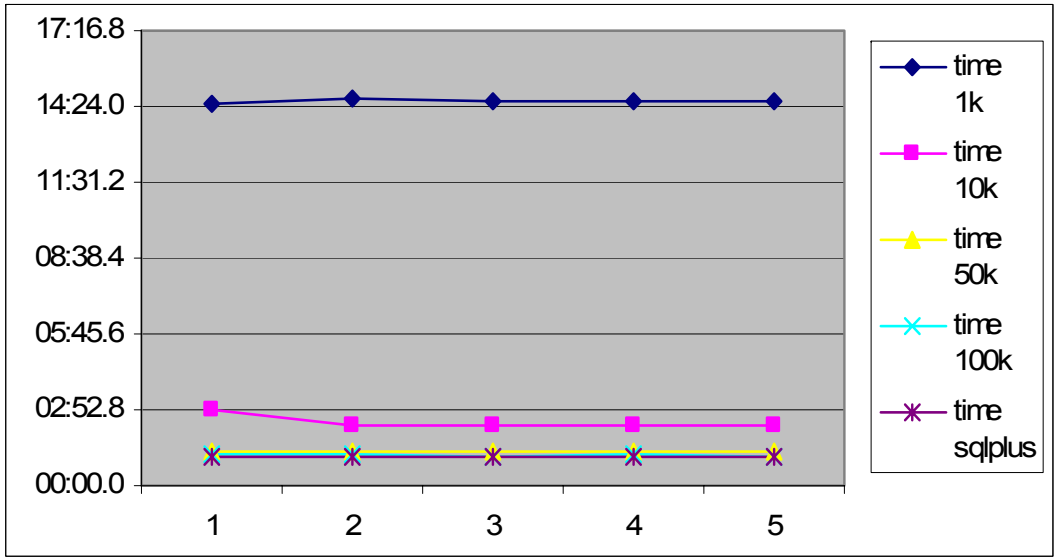


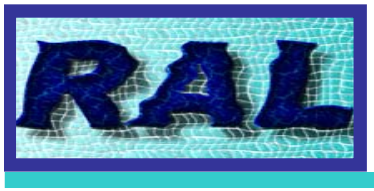


Relational File Catalog case 2 (Oracle)

- Query:
  pfn LIKE 'sfn://castorgrid.cern.ch/castor/cern.ch/cms/PCP04/%'
- Over 200000 rows in result set
- Oracle server side picture changing like weather in April
- 4-5% slower than SQLPLUS in the best case (100k prefetch buffer)

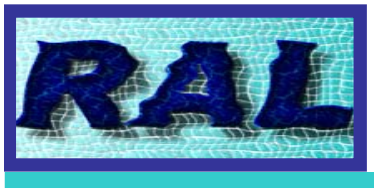






Recommendations for RAL clients

- Schema manipulation when using RAL
  - Try to avoid in production setups for building user schemas, use SQL scripts instead
  - RAL is very good in exploring database schemas but limited in creating optimal schemas
- Variable binding is not just a buzzword
- Don't try to fiddle around with AttributeList buffers
  - RAL really binds data addresses down to native API result sets and input buffers
  - Fill your application data using the AtrributeList items and do not copy them if possible
- Verify that your user schema has defined the indices where needed it often really helps
  - Use EXPLAIN SQL statement to see if you need one



New Developments

- Internal RAL review
  - Extensions for schema and table manipulations
  - Support of LOB types
  - Interface updates to simplify usage
  - More flexibility for client customization
- Turning RAL into separate LCG project
  - Separate CVS and release schedule
  - POOL becomes RAL client
  - Assumes pool::AttributeList being migrated into SEAL
- RAL client side monitoring
  - Cooperation with LCG 3D project and ATLAS
- MySQL 4.1.x native plug-in (backward compatible with 4.0.x)
  - Binary protocol & variable binding (big plus)
  - Still no cursors in 4.1 (emulated prefetching needed)
- ODBCAccess plug-in
  - code re-factoring (low priority)
  - Migration to the updated MyODBC driver 3.51.11 built against MySQL 4.1.x client
  - Possibly to MyODBC 3.53 when becomes available

Summary

- All back-ends heavy stressed by POOL ObjectRelational StorageService
  - Oracle plug-in works in all cases
  - SQLite & MySQL plug-ins in 99%
    - Issues related to the lack of server side cursor during nested queries
- RAL successfully used in implementationa across all POOL application domains
  - File catalog, Collections, StorageService
- RAL usage in COOL conditions database project is rumping up
  - Very useful feedback from stability and performance point of view
- Our Thanks to CMS and ATLAS developers for close collaboration and useful feedback

Issues

- Nested queries problems with ObjectRelational StorageService
  - SQLite & MySQL/ODBC (under investigation)
- CLOB trap when using bulk inserts
  - '\0' bytes not truncated by MySQL for TEXT columns
  - to be fixed in MySQL & checked for Oracle plug-in
- MySQL 4.0.x InnoDB does not scale well over 10^6 entries
  - Perhaps due to single shared table space file
  - We'll see in 4.1.7 where table space-per-table is possible
  - TEXT column type to be used with care
    - Storage overhead + slow query speed

POOL components