

Discussion of COOL - CORAL - POOL priorities for ATLAS

Andrea Valassi (IT-ES)

For the Persistency Framework team

ATLAS Database Meeting, 18th April 2011

- **Work in progress (feedback?)**
 - CORAL network glitch issues [RT, AK, AV]
 - Including fixes for ST / MT crashes [AV, RT, AK]: OK for Oracle. to be fixed in other 4 plugins
 - CORAL (and ~POOL) test infrastructure consolidation [AK, RT, AV]
 - COOL transaction cleanup [MW – committed, under review/integration AV, to be fixed]
 - PF port to cmake [SR, RT – plus SPI]
 - PF analyze/fix Coverity defects [MW for COOL?]
 - CORAL queries on two schemas [AK – committed, to be fixed, CMS request]
 - CORAL R&D, e.g. compare Frontier/CoralServer, SSD [AL – suggestions?]
- **Patches to be reviewed and/or released (when?)**
 - COOL vector payload a la CoraCool [MW – committed, to be released (ABI change)]
 - COOL expose sessions/transactions [MW – not committed yet (ABI change)]
 - COOL more efficient date storage [MW – not committed yet (ABI and schema change)]
 - CORAL sequences [AK – request A. Salnikov – committed, to be reviewed/released (ABI change)]
- **Work not (any longer) in progress (feedback?)**
 - CORAL support for FKs in SQLite [AK – committed, to be fixed, CMS request]
 - CORAL and CoralServer monitoring [AK prototype – committed, to be reviewed]
 - CORAL server secure authentication SSL + VOMS [MW – committed, to be reviewed/released]
 - CORAL server read/write functionalities [no progress – offline? online request G. Lehmann?]
 - CORAL size of simple Attribute [no progress – CMS request]
 - CORAL and COOL partitioning [many tests done, no code changes]
- **New fixes and enhancements (priorities? who?)**
 - PyCOOL faster BLOB access [MW? MC? – request J. Stelzer]; also: iterating slow in PyCool?
 - COOL API to get first/last IOV in tag [MW? – request A. Formica]
 - CORAL API to check if Attribute exists [MW? – request MW and CMS (ABI change)]
 - Review CORAL & COOL caches (data dictionary queries, transactions...)
 - CORAL performance: avoid data dictionary queries (schema cache) if not strictly needed?
 - CORAL API to cleanly distinguish serializable and non-serializable RO transactions?
 - COOL tags table cache; proper handling of non-serializable RO transactions?
 - Others: PF move to SVN? [MC, SR?]; PF valgrind analysis; PF documentation...

- **CORAL and COOL new features trigger API changes**
 - e.g. CoraCool replacement, sequences...
 - API changes lead to non-binary compatible libraries (ABI change)
- **ATLAS HLT uses pieces of online and offline software?**
 - How are new offline releases used in the HLT?
 - If we change the CORAL API, do you rebuild the ATLAS (online and offline) code that uses CORAL (against the new CORAL headers) or do you expect that the CORAL API/ABI is the same?
 - What needs to remain binary compatible?
 - If we publish a new CORAL, do you pick it up as a prebuilt library (with our ABI and our Boost externals for offline) for HLT, or do/can you rebuild it (so that potentially we can disable the ABI changes and use the same Boost)?
 - If the only way out is keep binary compatibility, when is a good time for the CORAL/COOL API changes to be released?
 - We are now holding back to keep the same APIs (for more than 12 months)
 - Apologies for maybe missing a window of opportunity at the start of 2011... ☹



- **CORAL data dictionary queries**
 - e.g. before querying MYTABLE, check if it exists
 - is this needed? (e.g. proper escaping or lowercase)? performance penalty?
 - info is cached inside Schema objects (all tables/views in a schema)
 - this is linked to Transactions (relies on serializable R/O? buggy for RW?)
 - buggy anyway (ok for nominal schema, not for others)
- **CORAL transactions**
 - R/W for writing (NB: DDL is auto-committed anyway)
 - R/O is serializable by default in CORAL
 - start transaction at t0; someone else updates data at t1; at t2 I see the state at t0
 - can be a nice feature to cache data (I check at t0, I do not need to check later)
 - it is a pain for the database server (need redo logs, can fail with ORA-01555)
 - another type: non-serializable R/O (e.g. default sqlplus)
 - needed by CoralServer in HLT (config data updated while transaction is on)
 - hacked in CORAL with an env variable (add proper API?); not handled in COOL
 - is this what Frontier does by default? is this ok for all usages?
- **COOL in-memory caches**
 - No caching in update mode
 - Caching in R/O mode (expect serializable) – one transaction per session
 - e.g. cache folder table; add caching of tag table?
 - add handling of non-serializable R/O (no cache)?



Reserve slides

- **Different issues reported by all experiments**
 - e.g. ORA-24327 “need explicit attach” in ATLAS/CMS ([bug #58522](#))
 - Fixed with a workaround in CORAL 2.3.13 (released in LCG 59b)
 - e.g. **OracleAccess crash** after losing session in LHCb ([bug #73334](#))
 - Fixed in current CORAL 2.3.16 candidate (see below)
- **Work in progress since a few months** (*A.Kalkhof, R.Trentadue, A.V.*)
 - Catalogued different scenarios and prepared tests for each of them
 - Prototyped implementation changes in ConnectionSvc and plugins
- **Current priority: fix crashes when using a stale session**
 - May be caused both by network glitch and user code ([bug #73834](#))!
 - A major internal reengineering of the plugins is needed (replace references to SessionProperties by shared pointers)
 - Done for OracleAccess ST in 2.3.16 candidate, pending for other plugins
 - The patch fixes single-thread issues; MT issues are still being analyzed
- **Next: address actual reconnections on network glitches**
 - e.g. non serializable R/O transaction: should reconnect and restart it
 - e.g. DDL not committed in update transaction: cannot do anything



- **Address recent bugs reported by the experiments**
 - e.g. performance penalty from COOL exceptions ([bug #79937](#))
- **API extensions to CORAL and COOL in the pipeline**
 - e.g. many payload per IOV, replace ATLAS CoraCool ([task #10335](#))
 - This breaks binary compatibility to previous releases
 - **When will be a good time for ATLAS (offline/online) to release this?**
 - Reviewing other pending requests for enhancements to plan ahead
 - **What are the highest priorities from ATLAS (e.g. in COOL)?**
- **CORAL performance studies and optimizations**
 - e.g. comparison of Frontier and CORAL server
 - e.g. studies of solid state disks with IT-DB
 - **Is there any other R&D that could be useful for ATLAS?**
- **Prepare future changes in the development infrastructure**
 - Port from CMT to cmake (with SPI and LHCb)
 - Move out of CVS eventually (to start with, of “LCG CVS”)

