



POOL/RLS Experience



Current CMS Data Challenges shows clear problems wrt to the use of RLS

- Partially due to the normal "learning curve" on all sides in using a new systems
- Some reasons are
 - Not yet fully optimised service
 - Inefficient use of the query facilities
- POOL and RLS service people works closely with production teams to understand their issues
 - Which queries are needed?
 - How to structure the meta data?
 - Which catalog interface?
 - Which indices?



More POOL/RLS Experience



- But poor performance also due to known RLS design problems!
- File names and related meta data are used for queries
 - Current RLS split of mapping data from file meta data (LRC vs. RMC) results in rather poor performance for combined queries
 - Forces the applications (eg POOL) to perform large joins on the client side rather than fully exploit the database backend
- Many catalog operations are bulk operations
 - Current RLS interface is very low level and results in large overheads on bulk operations (too many network round-trips)
- Transaction support would greatly simplify the deployment
 - A partially successful bulk insert/update requires recovery "by hand"
- These are not really special requirements imposed by POOL
 - Still acceptable performance and scalability needs a catalog design which keeps the data which is used in one query close to each other
 - Try to work around some of this know issues on the POOL side



Summary



- POOL Focus for 2004
 - Consolidation and Optimisation
 - RDBMS vendor independence
 - Common model for distributed, heterogeneous meta data catalogs
 - ConditionsDB production release and integration with POOL
- POOL will be a major ARDA/EGEE client
 - Needs to stay aligned with ARDA concepts and EGEE services
 - Provider of persistent object storage and collections
- Joint work package between POOL and ARDA in particular in the Collections area
 - Need more active experiment involvement
- Gaining valuable real life (data challenge) experience with POOL/RLS as input for next round
 - Produces concrete experiment requirements as input to ARDA/EGEE
 - POOL may be able to workaround some of the RLS design problems
- A real solution will be required from ARDA/EGEE to achieve the performance and scalability goals



Input for a next software generation



- Catalogs of “things” annotated with their meta data exist all over the system
 - These catalogs services could/should share the implementation and distribution mechanism
- Separation of catalog mapping data from associated meta data makes meta data almost useless
 - Efficient queries require that mapping and meta data are handled by (in!) one same database backend
- Higher level interface for bulk insert and bulk query is required
 - The current use of SOAP RPC call for each individual data entry will not scale to larger productions
- Transaction concept is required for a maintainable stable production environment
 - User transactions may span span several services!