Database Technical Evolution Group

Dario Barberis & Dave Dykstra

Report section editors: Rainer Bartoldus, Luca Canali, Gancho Dimitrov, Giacomo Govi, Mario Lassnig, Simon Metson, Andrea Valassi

Mandate & Operation model

- The Database TEG has been charged with addressing four major topics:
  - NOSQL vs Oracle, MySQL, etc
  - Frontier, squids, vs 3D/Streams, GoldenGate, DataGuard etc
  - CORAL, COOL
  - What is framework for deploying new applications that need a DB?
    - (e.g. if dev has used a random DB, how is that made into a service?)
  - There may be more topics to add as we go along.
- Timeline:
  - Workshop at CERN on 7-9 November 2011
  - Appointed section editors
  - Report to the GDB on 13 December 2011
  - Circulation of skeleton of report in December 2011
  - Addition of material to the report and e-mail discussions in January 2012
  - Report to WLCG (today)
  - Finalization of the report by end February

Databases for Conditions Data (1)

- Alice stores conditions data in ROOT files — non considered in the DB TEG context
- ATLAS, CMS and LHCb store conditions data in Oracle databases
  - ATLAS and LHCb use COOL, developed as a common project by the LCG Applications Area
  - CMS uses their own Conditions Database
  - Both solutions are based on CORAL
- Both COOL and the CMS Conditions Database are based on the concept of "Interval of Validity" (IoV) for each data group
- Both models allow the management of "tags" to distinguish between the multiple versions of a certain item of payload data in a given time range
- COOL models individual data payloads as relational records, i.e. arrays of simple strings, numbers and BLOBs, while the CMS software includes an object-relational layer (ORA) mapping the contents of tables to C++ objects
- CMS retrieves the full set of time ranges associated to a tag via simple SQL queries (not involving joins) and only looks up the time range for a specific event in C++ memory, while in COOL the selection of the appropriate conditions data for a given event time may be done at the SQL level
  - COOL SQL queries involving tags may be quite complex

Databases for Conditions Data (2)

- ATLAS has a much larger conditions database compared to both CMS and LHCb
  - ATLAS: 1.2 TB in Oracle, including indices and increasing by 0.7 TB per year
  - CMS: 200 GB in Oracle, increasing by 20 GB per year
  - LHCb: 2 GB when exported to SQLite, including indices and increasing by 0.6 GB per year
  - In addition, part of the payload is stored in external files, referenced in COOL, both in LHCb (magnetic field maps) and in ATLAS (calibration files)
- ATLAS started an internal review of conditions data handling aiming at
  - Improving performance by reducing the number of DB accesses per job and (possibly) memory occupancy
  - Rationalizing its use of COOL to use a more consistent approach across subdetectors and determine which of the many COOL features can (now or eventually) be dropped
- ATLAS calibration files are now available through CVMFS (much more reliable than Grid SRM); direct integration in the DB to be evaluated for 2015
- LHCb is now distributing SQLite replicas and evaluating using Frontier for remote DB access
- No experiment is planning to revisit substantially the Conditions Database model and the support tools (lack of specialized manpower)
  - Adiabatic improvements are always in the pipeline

Databases for Conditions Data (3)

- CORAL (ATLAS, CMS, LHCb) and COOL (ATLAS, LHCb) are examples of very successful common projects
- The CMS conditions software relies on CORAL for accessing Oracle (both directly and via Frontier) and SQLite.
  - CMS expects that CERN IT will provide the users with an adequate support for the CORAL framework, which should be devoted mainly in bug fixing and in improving performance bottlenecks when identified.
- ATLAS relies on COOL for the conditions database infrastructure, on CORAL to access the database layer, and on CoralServer for database access in the online environment.
  - ATLAS expects that these products will be supported by CERN as long as they are used by ATLAS and other experiments.
- LHCb needs CORAL and COOL to continue to be supported, as they are essential components of the LHCb software.

Access to Conditions Data

- CMS, ATLAS and soon LHCb use Frontier/Squid for Conditions DB access.
  - Central monitoring of worldwide Squids is very important to keep the Squids operating properly. The monitoring is now done by computers operated by CMS Frontier, but we recommend that a plan be made to transition the Squid central monitoring to WLCG
  - Locating the Squids is currently done separately per experiment and application, but we recommend that there be a WLCG standard way for jobs to locate Squids
  - We recommend that sites share Squids for all production services (currently Frontier and CVMFS)
  - Frontier/Squid should be recognized as a WLCG service and treated accordingly (GOCDB, GGUS, central rpm repository, monitoring)
- CMS and ATLAS differ on their approach to Conditions DB and Frontier servers
  - CMS currently has them only at CERN and is looking to add a second copy at a Tier 1 for improved resilience and reduced latency of cache-filling
  - ATLAS currently has them at CERN and 5 Tier-1s
    - ATLAS will reevaluate the cost/benefit of having more than one copy outside CERN during the 2013 shutdown

Oracle Operation Issues (1)

- Oracle services for WLCG have followed since their beginning the idea of providing high availability and scalability while using commodity components.
- Current production deployment is based on the following main building blocks:
  - Oracle clusters of commodity hardware
    - Clusters of 2 to 6 machines, depending on application load.
  - Shared storage compatible with RAC
    - NAS storage on Netapp over 10GE used at CERN for production since Q1 2012.
  - Oracle database software with RAC (real application clusters) option
    - Upgrade to Oracle 11gR2 in Q1 2012, previously 10gR2.
  - Database servers installed with RedHat enterprise Linux
- This type of architecture, based on clusters of hardware and specialized software from Oracle has allowed databases services to profit from:
  - Consolidation: for the largest experiments dozens of applications are consolidated in a single cluster. This reduces the maintenance effort.
  - High availability: systems are resilient against failures of individual components, notably failure of server nodes.
  - Homogeneous architecture: clusters of different sizes are built from a common base of building blocks of servers and storage. This simplifies configuration and reduces effort.
  - Rolling upgrades: several types of software patches, notably the quarterly security patches and OS patches, can be applied in a rolling way with minimal disruption of the services.

Oracle Operation Issues (2)

- Oracle software offers a large range of features for database processing that come at the cost of complexity. New software releases come out at regular intervals and provide security patches, bug fixes, major production versions. Current operational experience has shown that this reflects in the following critical areas:
  - Need for extensive testing by the application owners and database administrators (DBAs) for all changes
  - Need of expert knowledge for configuration, tuning and troubleshooting
- Following a solid validation procedure is often time consuming and costly, although crucial for smooth production deployment. In particular, experience has shown that the following practices should be followed:
  - Testing under load: this is to avoid the case where applications work fine in integration and may break in production with a larger users base
  - Performance testing against production-like data: i.e. SQL executed against small subsets of production data in test may not show critical performance and scalability issues
  - Changes should be approved by the relevant stakeholders and timely announced
  - Database applications need to be written with solid level of logging instrumentation. This has proven to dramatically reduce the effort needed for troubleshooting
- The most complex issues regarding the interaction of database applications and Oracle database services often lay at the boundary of competence between developers and DBAs. It is often the case that solutions have to be found case by case and also may vary in time. It is therefore very important to have a very good communication flow between the developers and DBA team.

Oracle Operation Issues (3)

- Some other challenges that have emerged are linked with the discussion on server management, data archiving and replication, and include:
  - Ensure correct capacity planning. Provide timely hardware upgrades when needed, for example with faster and more CPUs, more memory, SSD cache on the storage.
  - Newer Oracle database versions to fix some of the current issues and offer more functionality to explore and use.
  - Decouple transactional and non-transactional mode (for example with the use the Oracle 11g Active Data Guard and/or ad-hoc reporting databases)
  - Exporting older data to an archive system
- In all these activities the close co-operation of application developers, experiment DBAs and CERN-IT DBAs is essential for the deployment of new applications and the smoothness of database operations

Oracle Operation Issues (4)

- Database replication technology provides solutions to some very important use cases for WLCG databases:
  - Replication of conditions from online to offline databases,
  - Replication of PVSS data from online to offline,
  - Replication of ATLAS and LHCb conditions to some Tier-1 sites,
  - Replication of LFC for LHCb.
  - A new use case in 2012 is the replication of the ATLAS consolidated LFC data to BNL.
- Oracle Streams has been the technology of reference for replication of experiments data since the start-up of the database services.
- Oracle Active Data Guard is a technology that promises more robustness and performance and seems very much suitable for online to offline replication.
- Oracle Streams version 11g has shown in testing to have considerable improvements over its 10g version concerning critical areas such as performance and manageability. Streams has the additional flexibility over Data Guard of being able to replicate subsets of the source database in a more natural way.
- During 2012 Active Data Guard is planned to be introduced, first to complement, then to replace Streams for online to offline replication for CMS and Alice. Active Data Guard will also be used to replicate data to the stand-by databases, thereby enabling the use of a "free" read-only copy of all data.
- Later on the experience gained with Active Data Guard and Streams 11g in 2012 will allow taking decisions on the evolution of the architecture for the rest of the replication set-up both at CERN and at Tier-1 sites.

Production and Data Mgt Databases (1)

- Central databases for Distributed Computing applications have so far been based mostly, but not entirely, on Oracle
- During the last couple of years a few groups started exploring the NoSQL(*) databases that have recently appeared on the market

(*) NoSQL: a subset of structured storage systems, with hierarchical key-value pairs instead of rows & columns, and no "joins" required for complex yet flexible data structures.

They sacrifice some Relational DB features to improve performance for specific use cases, scale horizontally, and build resilience with cheap commodity hardware.

Production and Data Mgt Databases (2)

- The ATLAS DDM (Data Management) team investigated many NoSQL options and found that Hadoop/Hbase is a very good solution for the accounting tools
  - A "private" Hadoop cluster of 12 nodes is now in production; data are imported daily from Oracle production DB
  - Accounting tools: 8 hours in Oracle, 25 mins in Hadoop/SLC5, 6 mins in Hadoop/SLC6
    - Comparing an Oracle application that was optimized over the course of one year together with IT-DB with a straightforward one-week Hadoop implementation
    - But Oracle application running on a busy cluster vs Hadoop on dedicated h/w
    - Hadoop MapReduce vs Oracle SQL
- Similarly, the ATLAS Panda (Distributed Production and Analysis) and TDAQ teams found Cassandra useful for storing operations logs and running monitoring applications
  - Both under test
- CMS has been using CouchDB and MongoDB in production tools on VOboxes managed jointly by CMS and CERN-IT.
- Deploying and running these services has not been a burden to the operations team, in part because the scope of these tools was defined appropriately, and their deployment is considered a "quite successful scale-out experience" by the operators

NoSQL Remarks

- There is little in the way of "best practice" for developers, and internal training of developers has been necessary. We see this as a general cost with adopting a new technology, and not specific to the NoSQL tools.
- The learning experience for all tools has been smooth and relatively painless, both from a development and operations stand point.
- Hadoop has proven to be a stable, reliable, secure, and efficient platform for ATLAS OLAP workloads. Experience with Hadoop in production until now has shown that overall operational overhead is negligible, while providing efficient access to data and immense reduction of development and analysis time.
- Cassandra has proven to be a stable, reliable, secure and efficient platform for high throughput workload and random data access. Due to its high performance, it can be used in time-critical applications with minimal hardware and software effort.
- MongoDB has been used successfully in CMS for data aggregation but has been found to be inappropriate for ATLAS use cases, because it requires explicit partitioning of the data.
- CouchDB has been found to be a good fit for certain use cases in CMS, especially those related to monitoring (CouchDB can serve as both database and application platform) where replication between offsite databases and CERN is necessary. CouchDB has a RESTful protocol and read operations are potentially cacheable in Squid.

NoSQL Recommendations

- It seems to us evident that there are valid use cases for providing and supporting at least one of the NoSQL technologies at CERN
- In order to be able to properly advise developers within the experiments groups, the CERN-IT-DB group should test the most common NoSQL products for the already known use cases and acquire expertise with them
  - Technology tracking and market survey should also be part of this task.
- CERN IT should deploy a suitably sized Hadoop cluster
  - Focus on Hadoop rather than fragment effort over a variety of NoSQL tools
    - Other tools can, and will, be run ad-hoc by experiments as necessary
  - Hadoop clients, including pig/hive available on user interfaces (lxplus?)
  - Reasonably sized HBase installation
  - We make no operational requirements on the cluster, and appreciate that it will require training etc. for ops staff, so may run at low service level initially.
  - In the end it may need development, integration and production clusters
- Experiments would like to be involved in deployment discussions
- Build a community around the tools
  - Best practice doesn't really exist at CERN; have a forum to communicate what is learnt
  - Other groups may be interested in using these tools (Dashboard seems like a good candidate for example) but a central service is needed before expanding the user base

Next steps

- Complete and harmonize the report
  - Especially the section on NoSQL
- Submit to WLCG MB and GDB
  - Aim for 1 or 2 weeks following the discussion here
- What else?