Scaling up ATLAS Database Release Technology for the LHC Long Run

M Borodin\(^1\), P Nevski\(^2\), A Vaniachine\(^4\) on behalf of the ATLAS Collaboration

\(^1\) Moscow Engineering Physics Institute, Kashirskoe sh. 31, Moscow, 115409, Russia
\(^2\) CERN, CH-1211 Genève 23, Switzerland
\(^3\) Brookhaven National Laboratory, Upton, NY 11973 USA
\(^4\) Argonne National Laboratory, 9700 S Cass Ave, Argonne, IL 60439 USA

E-mail: vanachine@anl.gov

Abstract. To overcome scalability limitations in database access on the Grid, ATLAS introduced the Database Release technology replicating databases in files. For years Database Release technology assured scalable database access for Monte Carlo production on the Grid. Since previous CHEP, Database Release technology was used successfully in ATLAS data reprocessing on the Grid. Frozen Conditions DB snapshot guarantees reproducibility and transactional consistency isolating Grid data processing tasks from continuous conditions updates at the “live” Oracle server. Database Release technology fully satisfies the requirements of ATLAS data reprocessing and Monte Carlo production. We parallelized the Database Release build workflow to avoid linear dependency of the build time on the length of LHC data-taking period. In recent data reprocessing campaigns the build time was reduced by an order of magnitude thanks to a proven master-worker architecture used in the Google MapReduce. We describe further Database Release optimizations scaling up the technology for the LHC long run.

1. ATLAS Conditions DB and data processing on the Grid

ATLAS Conditions DB technology is hybrid: our “database” contains both database-resident information and external data in separate “payload” files, which are referenced by the database-resident data. These files are mostly in a common LHC format called POOL and there are also files in a ROOT format.

2.1. Benefits of hybrid Conditions DB technology

A snapshot of the ATLAS Conditions DB statistics collected in preparation for the LHC long run demonstrates the database complexity. There were 29 active Oracle schemas with 7,954 tables having 761,845,364 rows in total. The data volume was divided between 0.5 TB in Oracle and 0.2 TB in external POOL/ROOT files. To process a 2 GB file with 1K raw events a typical reconstruction job makes less than 2K database queries to read more than 70 MB of database-resident data, with some jobs read additionally similar volumes of extra data. In addition, an order of magnitude more conditions data have to be delivered in “payload” files. Our hybrid Conditions DB technology offloads that task from the Oracle database. (Delivering ~0.7GB of conditions data in files offloads the Oracle because in case of the non-hybrid database-resident Conditions DB technology large data volumes would be read directly from the Oracle server.)
2.2. On-demand data access on the Grid

A key feature of the Athena framework of the ATLAS experiment is on-demand data access. ATLAS production system delivers the required data to every Grid data processing job. Implementing infrastructure for on-demand data access on the Grid is not trivial. It is relatively straightforward to implement on-demand access for the input event data, since these files are known in advance of the data reconstruction job. The production system delivers those files to the Worker Nodes on the Grid. It is also straightforward to implement on-demand access for the database-resident portion of the conditions data because this is the use case the relational databases are used for in the first place. In contrast, building the Grid infrastructure for delivery of the Conditions DB data in files is challenging. This is because the conditions “payload” files are not known before the data processing job starts. Figure 1 shows the architecture and technologies used to deliver Conditions DB data on the Grid.

ATLAS database-resident information exists in its entirety in Oracle but can be delivered in smaller slices of data using SQLite or via Frontier/Squid web data caching infrastructure. These additional technologies are used to assure scalability of database access on the Grid [1]. Without these technologies, robust database access on the Grid requires database servers’ capacities adequate for a peak demand [2]. We found that deployment of the Oracle servers’ capacities required for scalable direct database access for data reprocessing on the Grid is not practical. The chaotic nature of Grid computing increases fluctuations in database load—in 2004 we observed the daily fluctuations in the database server load that were fourteen times higher than purely statistical [3].

2. Database Release technology

In 2005, to overcome scalability limitations in direct database access on the Grid, ATLAS introduced the Database Release approach, which is conceptually similar to the Software Release distribution on the Grid (Figure 2). The approach takes full advantage of technology independence in persistent data access, since no changes in the application code are required to switch between the technologies [1]. The ATLAS Conditions DB is partitioned in two instances: one for Monte Carlo, another for real data [4]. Depending on the type of input files, one of the Conditions DB instances is accessed by the Athena jobs. For Monte Carlo simulations, the Database Release integrates all Conditions DB data in a single tar file:

- a full replica of the Conditions DB instance for Monte Carlo in the SQLite file,
- a corresponding subset of Conditions DB POOL files,
- the POOL File Catalogue required for access to the Conditions DB POOL files.

In addition to the Conditions DB data, the Database Release includes SQLite replicas of the detector description and trigger configuration databases.
Years of experience resulted in continuous improvements in the Database Release approach, which now provides solid foundation for all ATLAS Monte Carlo simulation jobs [1]. For the jobs on real data use of the full SQLite replica is not practical, because of the Terabyte-scale size of the full Conditions DB instance for real data. For that reason, the Database Release for reprocessing contains only a necessary subset of the full Conditions DB data volume [5]. The Database Release technology fully satisfies ATLAS Computing Model requirements of data reprocessing and Monte Carlo production:

- fast data access: less than 10 s per job;
- robust: failure rate is less than $10^{-6}$ per job;
- scalable: served 1B queries in one of reprocessing campaigns;
- reproducibility of results: guaranteed by the read-only (“frozen”) replicas.

The Database Release technology assures transactional consistency by isolating data reprocessing jobs on the Grid from continuous conditions updates at the “live” Oracle server. Another benefit is the inherent partitioning by usage, which prevents accidental access to wrong conditions data [4].

2.1. Database Release build

Building a subset of the Conditions DB for a given run queries the Oracle database only once per run. During reprocessing thousands of jobs for a given run access only the SQLite replica, thus reducing Oracle load by many orders of magnitude (in comparison to the direct Oracle access in reprocessing).

Similar to the Software Release build, the Database Release preparation is on a critical path in the ATLAS reprocessing workflow. That is why it is important to reduce the time it takes to build and validate the Database Release (Figure 3). During the 2009 data reprocessing campaign the Database Release build had a completion time of about eight hours. The build has to be repeated due to several last-minute changes in the Conditions DB content, which delayed the overall ATLAS reprocessing workflow. Linear dependency of the build time on the number of runs chosen for the reprocessing indicated potential scalability bottleneck for the LHC long run. To scale up the Database Release technology for the LHC long run we minimized the Conditions DB Release build completion time using proven master-worker architecture similar to the Google MapReduce approach [6].

2.2. Inspirations from Google MapReduce

Our existing master-worker framework to parallelize tasks execution was presented at another CHEP Conference [7]. In contrast to Google MapReduce, where the master assigns each worker a task (the
“push” model); in our framework, when the worker task is short, the worker gets a next task from the master repository until all tasks are done (the “pull” model). The same “pull” model is used in the “pilot” job approach adopted by the LHC experiments [8].

A different approach to task parallelization – Google MapReduce – is inspired by Map and Reduce functions commonly used in functional programming. The Map and Reduce functions are both defined with respect to data structured in (Key, Value) pairs. The MapReduce steps can be nested.

Inspired by the MapReduce workflow, we extended our master-worker framework for parallel task processing adapting it for the Database Release build (Figure 4). Nesting and redundant execution extensions were "inspired" by the MapReduce steps. Since our input data reside not on the (scalable)
filesystem but in the database, we dynamically optimize the number of workers running in parallel to scale up our approach preventing potential Oracle server bottlenecks. Depending on the Oracle load from other concurrent tasks, the optimal number of workers is in the range of 10-50.

2.3. Avoiding scalability bottleneck during Database Release build
Application of the MapReduce ideas reduced the Conditions DB Release build time by an order of magnitude. It takes about 1.5 hours to build Conditions DB Release for the reprocessing campaign of the LHC data taken during December 2009–October 2010 period. The Database Release datasets compressed in 70 GB all conditions data needed to reprocess 150 runs corresponding to 65 days of continuous LHC data-taking.

2.4. Further Database Release optimizations
Further DB Release optimizations enabled operations in the open-ended mode, where the list of runs is not known in advance. Such mode of operations is necessary for the first-pass data processing on the Grid. The open-ended mode was successfully used during Heavy Ion run data processing on the Grid. Another advantage of the open-ended mode is the possibility to launch the reprocessing without waiting for Database Release build completion, which assures further scalability of the Database Release technology for the LHC long run.

3. Conclusions
Database Release technology fully satisfies the requirements of ATLAS data reprocessing and Monte Carlo production. Frozen Conditions DB snapshot guarantees reproducibility and transactional consistency isolating Grid data processing tasks from continuous conditions updates at the “live” Oracle server. To scale up the Database Release approach for the LHC long run we minimized the completion time of the Conditions DB Release build. Application of the MapReduce approach reduced the Database Release build time by an order of magnitude. Further optimizations added the open-ended mode of operations, which assures further scalability of the Database Release technology for the LHC long run by eliminating the need to wait for the completion of the Database Release build before the launch of the data reprocessing campaign. In addition, the open-ended Database Release build enables first-pass data processing on the Grid, which was used during Heavy Ion run.
Acknowledgments
We thank our collaborators for provided materials and discussions. We thank David Malon for presenting the talk on our behalf. This work is supported in part by the U.S. Department of Energy, Division of High Energy Physics, under Contract DE-AC02-06CH11357.

References