EPJ Web of Conferences **245**, 02006 (2020) https://doi.org/10.1051/epjconf/202024502006
CHEP 2019

ATLAS Tile Calorimeter Conditions Database architecture and operations in Run 2

Yuri Smirnov[1,*], *Dhiman* Chakraborty[1], *Alexander* Solodkov[2], and *Siarhei* Harkusha[3], on behalf of the ATLAS Collaboration[†]

[1]Northern Illinois University, Department of Physics, DeKalb, IL 60155, United States of America
[2]Institute for High Energy Physics of NRC Kurchatov Institute (IHEP), 142281 Protvino, Russia
[3]Institute of Physics of the NAS of Belarus, 220072 Minsk, Belarus

Abstract. An overview of the Conditions Database (DB) structure for the hadronic Tile Calorimeter (TileCal), one of the sub-systems of the ATLAS detector at LHC, is presented. ATLAS Conditions DB stores the data on the ORACLE backend, and the design and implementation have been developed using the COOL (Conditions Objects for LCG) software package as a common persistency solution for the storage and management of the conditions data. TileCal conditions and calibration data are stored in 4 separate Databases, each with its own schema: TileCal Online and Offline DBs for data, DB for Monte Carlo simulation and Detector Control System (DCS) DB. In order to ensure smooth operation of the TileCal during data taking, experts perform the necessary calibrations, add the changes of detector status and other conditions data, prepare new conditions for data reprocessing and Monte Carlo production campaigns, and upload the new up-to-date information into DB using custom-made software tools. The procedure of TileCal conditions' preparation, validation, uploading to DBs is described, and some DB-related statistics collected in Run 2 is presented.

1 Introduction

The Tile Calorimeter (TileCal) [1] is one of the sub-detectors of the ATLAS general purpose detector [2] operating at the Large Hadron Collider (LHC) at CERN. It is a large hadronic calorimeter which makes use of steel as the absorber material and scintillating plates read out by wavelength shifting (WLS) fibres as the active medium. It is crucial in identification of hadronic jets and measurement of their energy and direction. It also provides information for triggers and participates in the measurement of the missing transverse momentum carried by non-interacting particles. The TileCal consists of a cylindrical structure with an inner radius of 2280 mm and an outer radius of 4230 mm. The layout of the calorimeter is shown in Figure 1. It is subdivided into a 5640 mm long central barrel, consisting of two partitions (LBA, LBC) for operational purposes, and two 2910 mm extended barrels (EBA, EBC). The scintillating tiles lie in the r-φ plane and span the width of the module in the φ direction[‡].

* Corresponding author: ismirnov@niu.edu

‡ ATLAS uses a right-handed coordinate system with its origin at the nominal interaction point (IP) in the centre of the detector and the z-axis along the beam pipe. The x-axis points from the IP to the centre of the LHC ring, and the y-axis points upward. Cylindrical coordinates (r, φ) are used in the transverse plane, φ being the azimuthal angle around the z-axis. The pseudorapidity is defined in terms of the polar angle θ as $\eta = -\ln \tan(\theta/2)$. The long central barrels cover the region $-1.0 < \eta < 1.0$, and the extended barrels cover the pseudorapidity region $0.8 < |\eta| < 1.7$.

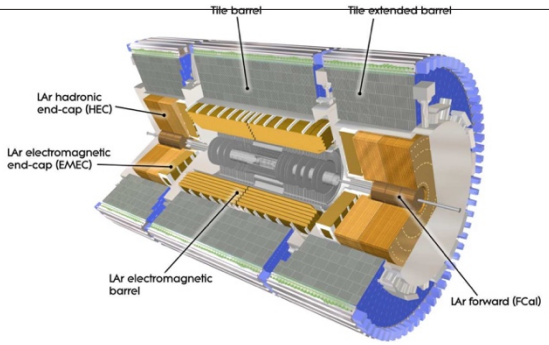

Fig. 1. Layout of ATLAS Calorimeter systems including TileCal (Tile) and Liquid Argon (LAr). Figure is taken from Ref. [2].

WLS fibres running radially collect the light from the tiles along their two open edges. Readout cells are then defined by grouping together a set of fibres into a photomultiplier (PMT), to obtain a three dimensional segmentation. Radially, the calorimeter is segmented into three layers, approximately 1.4, 3.9 and 1.8 interaction lengths thick at $\eta=0$; the $\Delta\eta \times \Delta\varphi$ segmentation is 0.1 x 0.1 (0.2 x 0.1 in the last radial layer, tail catcher). Each TileCal partition consists of 64 modules, and most of TileCal cells are read out by two PMTs, accounting for 9852 read-out channels in total corresponding to 5182 cells.

2 TileCal Conditions Database architecture

In Run 2 two ATLAS ORACLE Database instances have been used to store the conditions: CONDBR2 [3], keeping conditions for data (online data taking, offline data processing and DCS data), and OFLP200, keeping conditions for Monte Carlo (MC) simulation data production. Every ATLAS sub-system, including TileCal, uses several COOL [4] DB schemas. One of them for TileCal Offline Conditions DB is shown in Figure 2.

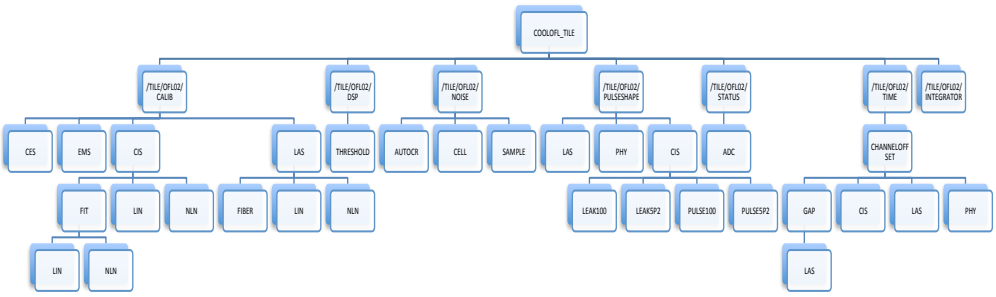

Fig. 2. Schema of COOL folders in TileCal Offline Conditions DB, CONDBR2 ORACLE instance.

Each schema in COOL is organized into a hierarchical folder structure, similar to a file system, and every leaf folder in the Offline DB keeps multiple versions of tags linked to ATLAS Global tags described in [3]. New leaf tags (and hence new ATLAS Global tag) with a new set of conditions data are usually created for every round of reprocessing. In particular, two different Global tags always exist: tag for the prompt reconstruction of express stream (10% of data), so-called UPD1 tag; and tag for the bulk reconstruction, so-called UPD4 tag.

In the Online DB the single (head) version tags are used, and the structure of folders in this DB is similar to the one presented in Figure 2. Every tag contains a set of conditions split into Intervals of Validity (IOVs) to allow for conditions to vary between runs and/or luminosity blocks within the same run. The luminosity block is the shortest time interval for which integrated luminosity can be determined.

To store the conditions in COOL DB we use Blob format. A Binary Large Object (Blob) is a collection of binary data stored as a single entity in a DB management system. TileCal conditions data are stored with modules granularity: for 4x64=256 TileCal modules we use 256 COOL channels containing one Blob each as a conditions payload. 64K Blobs are sufficient for the majority of conditions, but in a few special cases we use 16M Blobs. We apply offline module-hash [0-275] for indexing, using indices [0-19] to store partition depending default values. Such an approach allows us to reduce significantly (~20%) the DB volume and to avoid the duplication.

ATLAS Detector Control System DB in COOL is common for all subsystems and represents a special case. TileCal has only 3 folders in that DB for storing high voltage values, temperature, etc. Therefore, for simplicity and in accordance to uniform requirements for all subsystems, the payload is just an array of floats instead of Blobs here.

3 TileCal Conditions DB software

To communicate efficiently with TileCal Conditions DB (read, and write/update operations) both ATLAS-made DB tools such as AtlCoolConsole, AtlCoolCopy, AtlCoolMerge, COMA [5], etc. and the following TileCal-made specific software can be used:

- TileCalibBlobObjs – definitions of TileCal data types in COOL
- TileCalBlobPython – python methods to work in COOL
- TileConditions – package to work with IOVs
- TileCalibAlgs, TUCS – client tools for calibration
- TileMonitoring, Tile-in-One – tools for monitoring, calibrations, data quality [6]
- TileCalibWeb – web-based application for TileCal COOL DB updates
- TileDQ – software for TileCal data quality verification.

To see the structure and content of any TileCal Conditions DB the command line interface AtlCoolConsole.py, which is common to all ATLAS subsystems, can be used.

```
Connected to 'frontier://ATLF/C;schema=ATLAS_COOLOFL_TILE;dbname=CONDBR2'
Welcome to AtlCoolConsole. Type 'help' for instructions.
|>>> listtags /TILE/OFL02/STATUS/ADC
listing tags for folder /TILE/OFL02/STATUS/ADC
TileOfl02StatusAdc-EmptyBCh (unlocked) []
TileOfl02StatusAdc-RUN2-HLT-UPD1-00 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-08 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-09 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-10 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-11 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-12 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-13 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-14 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-15 (locked) []
TileOfl02StatusAdc-RUN2-UPD4-16 (unlocked) []
TileOfl02StatusAdc-RUN2-UPD4-17 (unlocked) []
TileOfl02StatusAdc-RUN2-UPD4-18 (unlocked) []
|>>> usetag TileOfl02StatusAdc-RUN2-UPD4-15
Changed current tag selection to TileOfl02StatusAdc-RUN2-UPD4-15
|>>> usechan 275
Changed current channel selection to 275
|>>> more /TILE/OFL02/STATUS/ADC
Using channel: 275
Using tag selection: TileOfl02StatusAdc-RUN2-UPD4-15
[215064,0] - [215091,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215091,0] - [215414,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215414,0] - [215415,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215415,0] - [215433,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215433,0] - [215464,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215464,0] - [215465,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215465,0] - [215473,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215473,0] - [215541,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215541,0] - [215542,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[215542,0] - [216399,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[216399,0] - [216417,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[216417,0] - [216590,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[216590,0] - [217512,1) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[217512,1] - [217543,1) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[217543,1] - [217906,1) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[217906,1] - [217949,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[217949,0] - [217999,1) (275) [TileCalibBlob (Blob64k) : size=596,chk=-660653637]
[217999,1] - [218048,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-1714499734]
[218048,0] - [218536,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-1714499734]
[218536,0] - [218589,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-1714499734]
[218589,0] - [218677,0) (275) [TileCalibBlob (Blob64k) : size=596,chk=-1714499734]
```

Fig. 3. Example structure of TileCal DB folder, with tags and IOVs.

In Figure 3 an example of TileCal Offline Run 2 Conditions DB on the AtlCoolConsole view is presented. It is easy to see all UPD1 and UPD4 tags for that folder, as well as a list of IOVs and payload Blob types used for various tags and 277 COOL channels.

We use two types of TileCal Conditions DB updates. A large DB update usually requires creation of the new tag needed for data reprocessing or for MC production campaigns and can be performed by uploading the corresponding SQLite file with help of AtlCoolMerge command executed from the command line. The second type of update is a minor correction of few COOL channels (for instance masking or unmasking the "bad" channels or cells) in TileCal. Such an operation can be done by adding a new IOV to the existing tag directly from the custom TileCalibWeb interface presented in Figure 4.

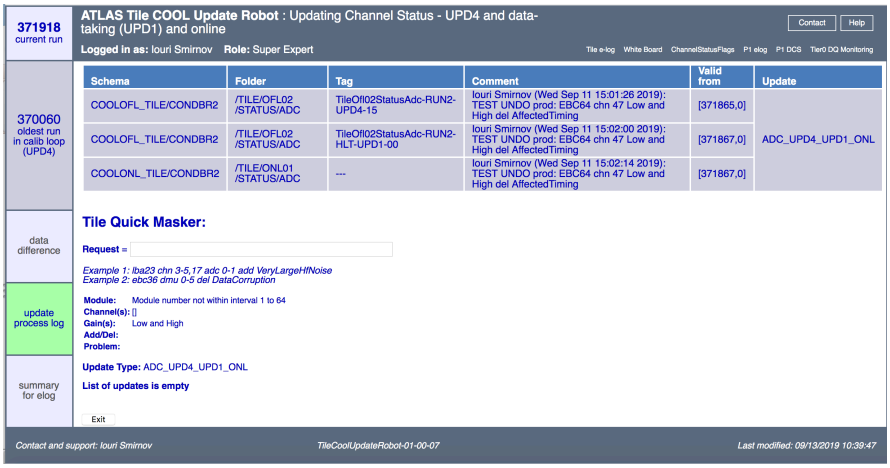

Fig. 4. TileCalibWeb (Robot) tool for Conditions DB updates.

To read the conditions back from COOL and display them we use python scripts (ReadCalibFromCool, PlotCalibFromCool, etc.) from the TileCalibBlobPython package. An example of plot showing the Charge Injection System (CIS) calibration constants (ADC counts / pC) for all the high gain channels in TileCal LBA modules for a single run taken in November 2018 is shown in Figure 5.

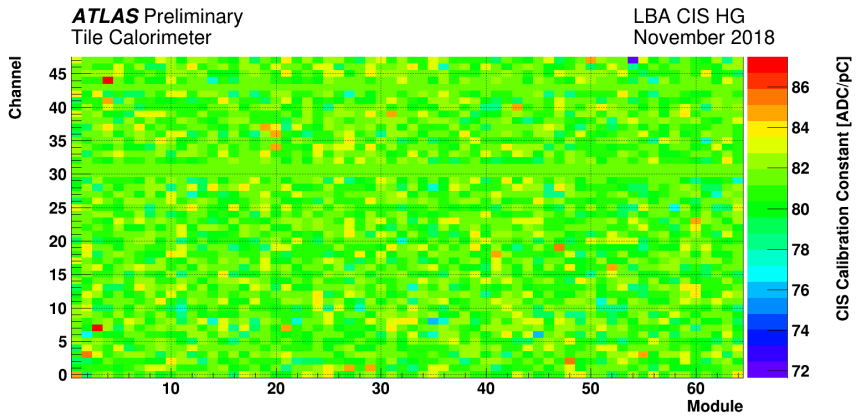

Fig. 5. Charge Injection System calibration constants for LBA modules, UPD4-18 tag. Figure is taken from Ref. [7].

4 TileCal Conditions DB operations during Run 2

TileCal Conditions Database operations include preparing, validating and performing frequent COOL DB updates, supporting the TileCal DB infrastructure (client tools and environment) and troubleshooting. The frequency of TileCal DB updates during the entire LHC Run 2 period (2015-2018) is shown in Figure 6. This statistics includes TileCal Online and Offline Conditions DBs both for data and MC simulations.

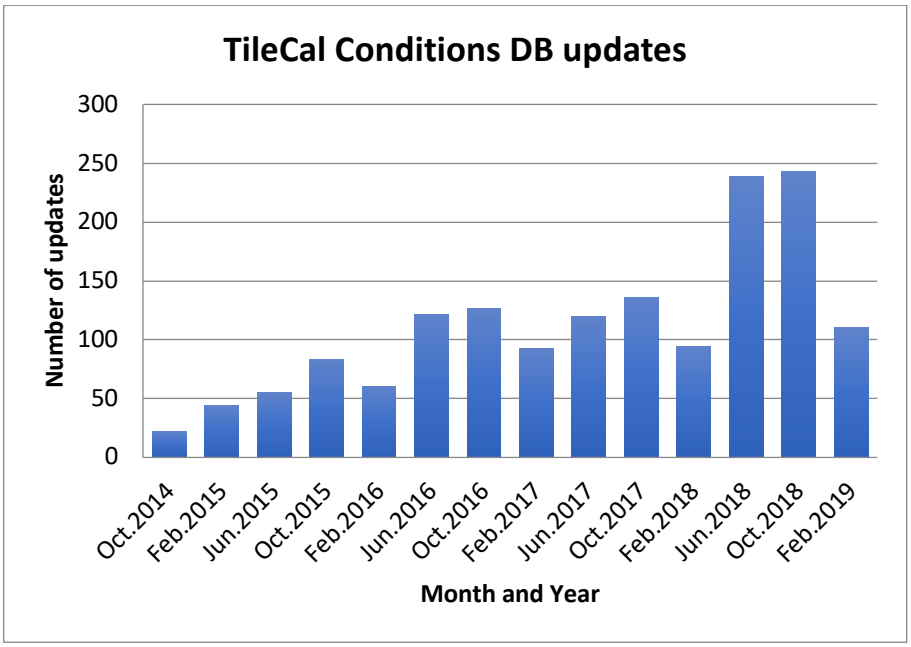

Fig. 6. TileCal Conditions DB updates summary.

Figure 7 shows the percentage of all ineffective cells and channels in the detector that are masked as a function of time starting from December 2010 till the end of Run 2 (3-rd of December 2018).

The hatched area represents the maintenance period of the detector. The legend includes the amount of masked cells (0.48%) and masked channels (1.07%) at that time. The cell is completely masked when both channels of this cell are masked, that's why the number of masked cells is smaller than number of masked channels. All operations, including these changes in the amount of masked channels, were performed through TileCalibWeb interface, which allowed us to produce DB updates very quickly, easily and efficiently.

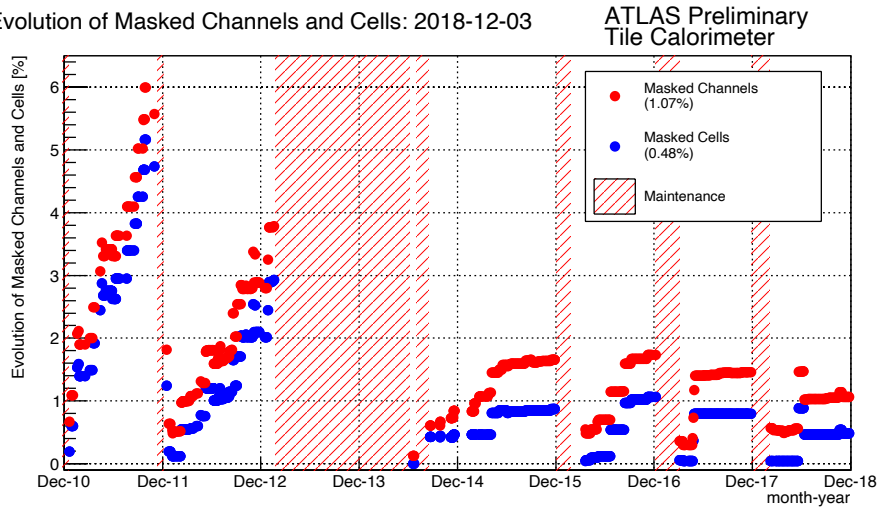

Fig. 7. Evolution of masked TileCal channels and cells. Figure is taken from Ref. [8].

Figure 7 also shows that the performance of TileCal was good and stable during the Run 2 period and much improved compared to Run 1.

5 Conclusion

During Run 2 operations of the ATLAS detector at LHC, the amount of TileCal Conditions DB transactions, in particular update operations, increased significantly. The TileCal DB architecture and software allow to optimize and automatize the procedure of conditions data preparation, validation and uploading into ORACLE COOL DB, and to provide the efficient support for smooth Run 2 TileCal operations and the data reprocessing.

References

1. The ATLAS Collaboration, *ATLAS Tile Calorimeter Technical Design Report*, CERN-LHCC-96-042162 (1996)
2. The ATLAS Collaboration, JINST **3**, S08003 (2008)
3. M. Böhler et al., J. Phys.: Conf. Ser. **664** 042005 (2015)
4. A.Valassi, M. Clemencic, D. Dykstra et al., J. Phys.: Conf. Ser. **331**, 042043 (2010)
5. E.J. Gallas et al., J. Phys.: Conf. Ser. **396** 052033 (2012)
6. A. Sivolella et al., J. Phys.: Conf. Ser. **664** 052034 (2015)
7. ATLAS Collaboration, *ATLAS Tile Calorimeter Public Plots*, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ApprovedPlotsTileCalibration#Charge_injection_system_CIS
8. ATLAS Collaboration, *ATLAS Tile Calorimeter Public Plots*, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ApprovedPlotsTileDetectorStatus