

# The Shuttle: The ALICE Framework for the Extraction of the Conditions Data

---

**Chiara Zampolli**<sup>\*†</sup>

*CERN and CNAF-INFN*

*E-mail: Chiara.Zampolli@cern.ch*

**Federico Carminati**

*CERN*

*E-mail: Federico.Carminati@cern.ch*

**Alberto Colla**

*University of Rome, La Sapienza, and INFN*

*E-mail: Alberto.Colla@roma1.infn.it*

ALICE (A Large Ion Collider Experiment) is the general-purpose experiment at the CERN Large Hadron Collider (LHC) designed for heavy ion (HI) physics.

ALICE will collect data at a rate of 1.25 GB/s during heavy ion runs, and at a rate of 100 MB/s during p-p data taking. In a standard data taking year, comprising of several months of p-p, one month of HI running and the associated Monte Carlo (MC) simulation, the expected total data volume is of the order of 2 PB. This includes raw data, reconstructed data (both Event Summary Data - ESD, and Analysis Object Data - AOD) and the calibration, environment and alignment data (conditions) necessary for the reconstruction and physics analysis.

The extraction and publication of the conditions data is handled by a system called Shuttle. Detector-specific routines, called preprocessors which are integrated in the Shuttle framework, process the data from online sources and produce the final conditions objects. The Shuttle itself provides access to the online conditions data sources, controls the execution of the preprocessors and publishes the conditions data on the Grid.

*13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research,*

*ACAT2010*

*February 22-27, 2010*

*Jaipur, India*

---

<sup>\*</sup>Speaker.

<sup>†</sup>for the ALICE Collaboration

## 1. Introduction

ALICE (A Large Ion Collider Experiment) [1] is the general-purpose experiment at the CERN [2] Large Hadron Collider (LHC) [3] designed for heavy ion physics. Mainly concerned with the study of Pb-Pb collisions, ALICE will also contribute to the LHC p-p program. During data taking, the 18 ALICE subdetectors interact with the five ALICE online systems: the DAQ (Data Acquisition), the DCS (Detector Control System), the ECS (Experiment Control System), the HLT (High Level Trigger) and the Trigger. The raw data produced are stored in the WLCG Grid [4], in order to be subsequently reconstructed. The conditions data necessary for the calibration and the alignment of the detectors as well as the environment parameters needed for the reconstruction should be extracted from the raw data themselves. This would imply an extra pass over the raw data before reconstruction, which is too expensive in terms of computing resources. In addition, not all the conditions data are stored in the raw data. For these reasons, a common framework dedicated to the extraction of the conditions data *during* data taking has been developed, named the Shuttle. The Shuttle handles and coordinates the ALICE detectors and online systems in the gathering, processing, and publication of the conditions data. The online systems and their databases are subject to stringent access control and have very limited outside exposure. The Shuttle provides an interface between the protected online world and the external computing resources. All collected conditions data are exported onto the Grid, thus making them accessible for the reconstruction and analysis tasks running on any of the Grid computing centers.

## 2. The Shuttle Framework

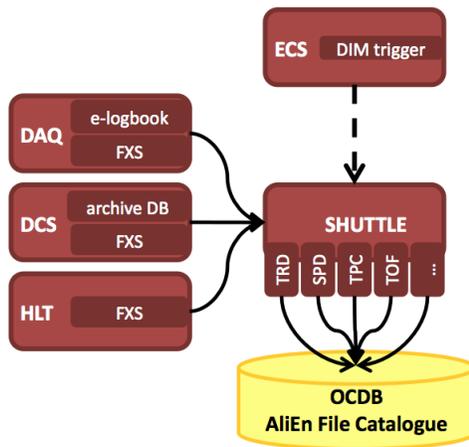
Being the only connection between the online systems and the offline world, the Shuttle accomplishes the necessary tasks to make the conditions data available to the detectors during reconstruction. The Shuttle conditions data handling process consists of copying the data produced by the online systems for each subdetector (in whatever format), preprocessing them (e.g. performing consolidation, fitting...), reformatting them in ROOT [5] files, and storing them in the Offline Condition Database (OCDB) in the Grid. The last step is done using dedicated AliRoot<sup>1</sup> [6, 7] classes that combine the ROOT functionalities and the LCG Grid technologies for the archiving, cataloguing and retrieval of the data in a Grid distributed environment. One should note that the OCDB is not a relational database, but a set of entries in the AliEn file catalogue, which point towards physical ROOT files stored in the various Storage Elements (SE) of the WLCG Grid.

The Shuttle sends signals to the online and Grid services once a run has been processed which means that the conditions data for that run are now available in the Grid, and that the run can be reconstructed. This is done inserting a dedicated entry in the AliEn file catalogue.

Figure 1 schematically shows the Shuttle program flow, starting from the online systems, going through the detector preprocessors, and finally transferring the data produced by the preprocessors to OCDB.

---

<sup>1</sup>AliRoot is the ROOT-based ALICE offline software for the simulation, the reconstruction and the analysis of data.



**Figure 1:** Schematic representation of the Shuttle framework.

### 3. The Shuttle Program Flow

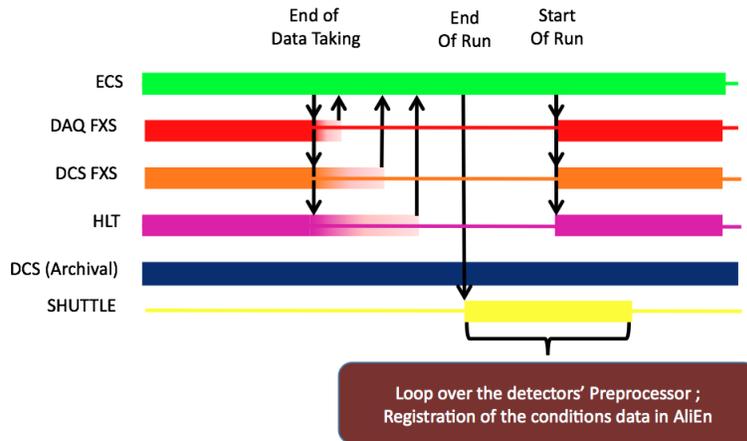
The Shuttle flow starts reading the configuration parameters (such as those necessary to access the ECS logbook where the run-specific information is kept) from an LDAP database. Then, the Shuttle receives the trigger from ECS at every end-of-run, after which it retrieves from the ECS logbook the run information that is relevant for its operation (such as the list of detectors that were active during that run). The next step consists in running the detector specific procedures, called *preprocessors*, in charge of extracting the conditions data needed for the reconstruction from the data collected by the online systems. Within these procedures, the Shuttle accesses the online systems (DAQ, DCS and HLT) to retrieve the conditions data they have produced for that run, in any format for every active detector. The Shuttle finally stores the conditions data produced by the preprocessors in the OCDB. In parallel, the data useful for a later cross-check of the results (the so-called Reference data) are stored in a second OCDB-like database.

It is important to underline that, even if the Shuttle program runs triggered by the ECS end-of-run signal, being asynchronous with respect to data taking, it does not interfere with it. New runs can be started whilst the Shuttle is still processing already finished ones. This is schematically shown in Figure 2 (see Sec. 3.1 and Sec. 3.2 for more details).

In general, the Shuttle program flow consists of two main steps: the gathering of the conditions data produced in the online systems, and the running of the detector-specific preprocessors.

#### 3.1 Interaction with the Online Systems

The three online systems DAQ, DCS and HLT can produce data for every subdetector, running dedicated programs (Detector Algorithms - DA) on their machines, and saving the output in special File Exchange Servers (FXS), from where the Shuttle can then ship them. As shown in Fig. 2, at the end of data taking the ECS informs the online systems that data taking has stopped. As a result, the DAs procedures complete the processing of the conditions data collected during the run. Once the results from the DAs have been stored in the FXS, the online systems send a signal back to



**Figure 2:** Schematic representation of the Shuttle program flow.

ECS. Finally, the ECS sends an EOR signal to the Shuttle, in order to trigger its operation detector per detector.

Another source of data for the Shuttle comes only from the DCS and consists in the detector-specific parameters (such as high voltages, temperatures, thresholds...) which are continuously monitored and archived in a dedicated Oracle database by DCS (see Fig. 2). Thanks to a specific server-client communication protocol (standard TCP/IP), the Shuttle can access the stored values to be then passed to the subdetectors.

### 3.2 The Preprocessors

The second key component of the Shuttle framework are the detector preprocessors. These procedures, which are the direct responsibility of the detectors and are fully integrated into AliRoot, are run independently and sequentially by the Shuttle, which, in turn, monitors them in terms of memory consumption and processing time. At every run, and for every preprocessor, the Shuttle program forks: the child process executes the current preprocessor, while the parent process executes the main Shuttle program monitoring the child. This assures that the crash of one preprocessor does not affect the main process, and that no memory leak can occur. It is during the execution of each preprocessor that the Shuttle accesses the data stored in the File Exchange Servers and in the DCS archive.

The status of each preprocessor at every run is monitored by the Shuttle in order to allow error recovery. The possible statuses for each preprocessor are shown in Fig. 3. The logic of the processing can also be inferred, from the start of the execution of each preprocessor (“Started”), to the successful or unsuccessful completion (“Done” or “Failed”). Moreover, the status changes of both the Shuttle and the preprocessors are sent to a MonALISA [8] server and visualized by the ALICE MonALISA repository. From here, the processing status, the history and the output of the processing can be accessed. For more details about the possible preprocessor statuses and about the MonALISA monitoring, see [9, 10, 11].

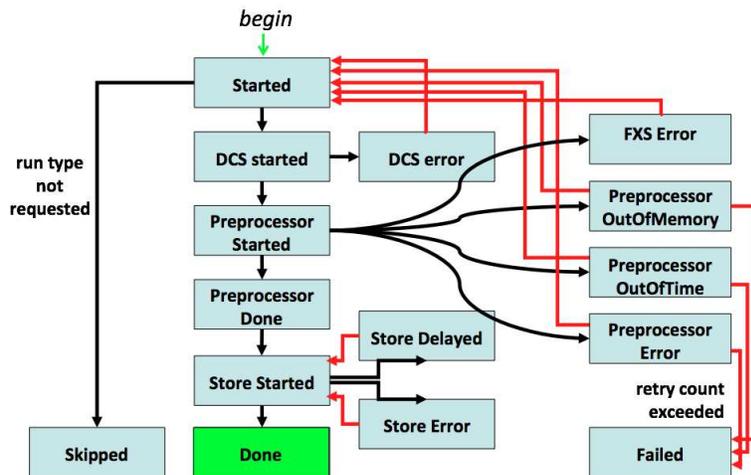


Figure 3: Preprocessor possible statuses.

#### 4. The Shuttle in Production

The Shuttle has been collecting ALICE conditions data since December 2007, when the first ALICE commissioning phase took place.

Since 2008, and until February 2010, about 67000 runs have been successfully processed, corresponding to about 180000 files in the DAQ FXS. About 85% of the total amount of runs taken by ALICE have been processed by the Shuttle. Of these, 20% of these were PHYSICS runs, while the remaining 80% were calibration runs. The corresponding conditions data have been stored in  $\sim 183000$  files in the AliEn file catalogue, for a total size of  $\sim 30$  GB. Moreover,  $\sim 8.4$  GB of reference data (see Sec. 3) have been stored in  $\sim 29000$  files in AliEn.

Finally, it is worth mentioning the role of the Shuttle framework within the quasi-online reconstruction steered from the ALICE Control Room during data taking. Some of the functionalities developed within the Shuttle schema are in fact used in order to retrieve the run parameters and the conditions data necessary to it.

#### 5. Summary

The ALICE framework for the extraction of the conditions data called Shuttle is the only link between the online systems DAQ, DCS, ECS, HLT and Trigger as well as the ALICE offline world. It is in charge of producing the conditions data necessary for the reconstruction, by running detector specific procedures called preprocessors. Since the first time it became operational, in December 2007, the framework has been going through various upgrades, while the development of new features within the preprocessors is done in a dedicated automatized nightly test setup, which reproduces the actual Shuttle operation flow. So far, the Shuttle has run smoothly during cosmics runs, calibration runs, and LHC data taking, being one of the main ingredients for ALICE data taking, before analysis can start.

## References

- [1] K. Aamodt et al. [ALICE Collaboration], *The ALICE experiment at the CERN LHC*, **JINST 3 S08002**, 2008
- [2] <http://user.web.cern.ch/user/Welcome.asp>
- [3] <http://lhc.web.cern.ch/lhc/>
- [4] <http://lcg.web.cern.ch/LCG/>
- [5] R. Brun, F. Rademakers, *ROOT: An object oriented data analysis framework*, **Nucl. Instrum. Meth. A 389 81**, 1997
- [6] ALICE Collaboration, *ALICE Technical Design Report of the Computing*, **CERN-LHCC-2005-018**, 1995
- [7] <http://aliceinfo.cern.ch/Offline/>
- [8] I.C. Legrand, H.B. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, M. Toarta and C. Dobre, *MonALISA: An agent based, dynamic service system to monitor, control and optimize grid base applications*, *Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP)*, Interlaken, Switzerland 27 Sep - 1 Oct 2004
- [9] A. Colla and J.F. Grosse-Oetringhaus, *The Shuttle Framework - A system for automatic readout and processing of conditions data*, *ALICE internal note*, **ALICE-INT-2008-011**, 2008, <https://edms.cern.ch/document/924807/1>
- [10] J.F. Grosse-Oetringhaus, *Measurement of the Charged-Particle Multiplicity in Proton-Proton Collisions with the ALICE Detector (Appendix D)*, 2009, *Ph.D. thesis, University of Münster*
- [11] J.F. Grosse-Oetringhaus, C. Zampolli, F. Carminati and A. Colla, *The ALICE online-offline framework for the extraction of conditions data*, *Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP)*, Prague, Czech Republic 21- 27 March 2009