

TOF CALIBRATION AND ALIGNMENT



the tunnel...

TOF Online Calibration: status

SHORT RECAP FROM JULY OFFLINE WEEK

- **Type 1 procedure:** Detector Algorithm: results processed at the EOR by the Shuttle, output used in reconstruction: *done*
- Using the decoding classes in AliROOT in the DA
- **Online calibration** already implemented and successfully tested:
 - DAQ/DCA/ESC/TRG/TOF integration test in April: *validation* of the TOF-DA in the DAQ environment;
 - Shuttle Preprocessor already implemented and tested;
- **Type 4 procedure:** monitoring of a set of DCS data points, storing summary information to reference data that are not used in reconstruction: *done*
- **Type 3 procedure:** dead channel map from pulse eventually noisy channels from online monitoring to be exported to DCS FXS: *to be done*

DISCUSSION HAS STARTED

TOF Online Calib: Type 3 procedure

- To define the status map of the channel in the OCDB (stored as 1 char/channel)
- Should result from the combination of 3 types of information:
 - **hardware** channel status (ON/OFF)
 - **pulser** test
 - **noise** test

Channel Status Map: Sources

- ~ **Hardware** channel status (ON/OFF):
 - ~ Should come from the info that the TOF DAQ and DCS export to the **DCS FES**; from there, AMORE should be able to read, to produce a map (1D histo) to be re-exported to the **DCS FES**, from where the Shuttle can finally ship it; (1)
- ~ **Pulser/Noise** tests: 2 possible ways:
 - ~ *Either* AMORE may produce two separate 1D histos, write them on the **DCS FES**, so that then the Shuttle can ship them; (1)
 - ~ *Or* a dedicated DA may run, export 1D histos to the **DAQ FES**, from where the Shuttle may ship them. (2)
- ~ **Pending questions:**
 - ~ (1) *Can AMORE export to the DCS FES?*
 - ~ (2) *Can the DAQ define a Pulser and a Noise run type for TOF (distinguishable within a calibration run)? ...under discussion...*

TOF Offline Calibration

- ~ On-the-Fly offline calibration *redesigned* wrt July Offline week, being splitted into *two steps*:
 - ~ **TTask** (post-reconstruction process):
 - ~ Filling a tree:
 - ~ one entry/TOF channel;
 - ~ Each entry being a 1D array with the necessary information to perform calibration;
 - ~ Running at the EOR;
 - ~ Writing the tree into AliEn (as reference data) → don't need CDB access
 - ~ **Job/Macro**:
 - ~ Chaining the trees so far created over many runs
 - ~ Running the calibration process
 - ~ Writing the calibration parameters on the CDB

TOF Offline Calibration: AliTOFCalibTask

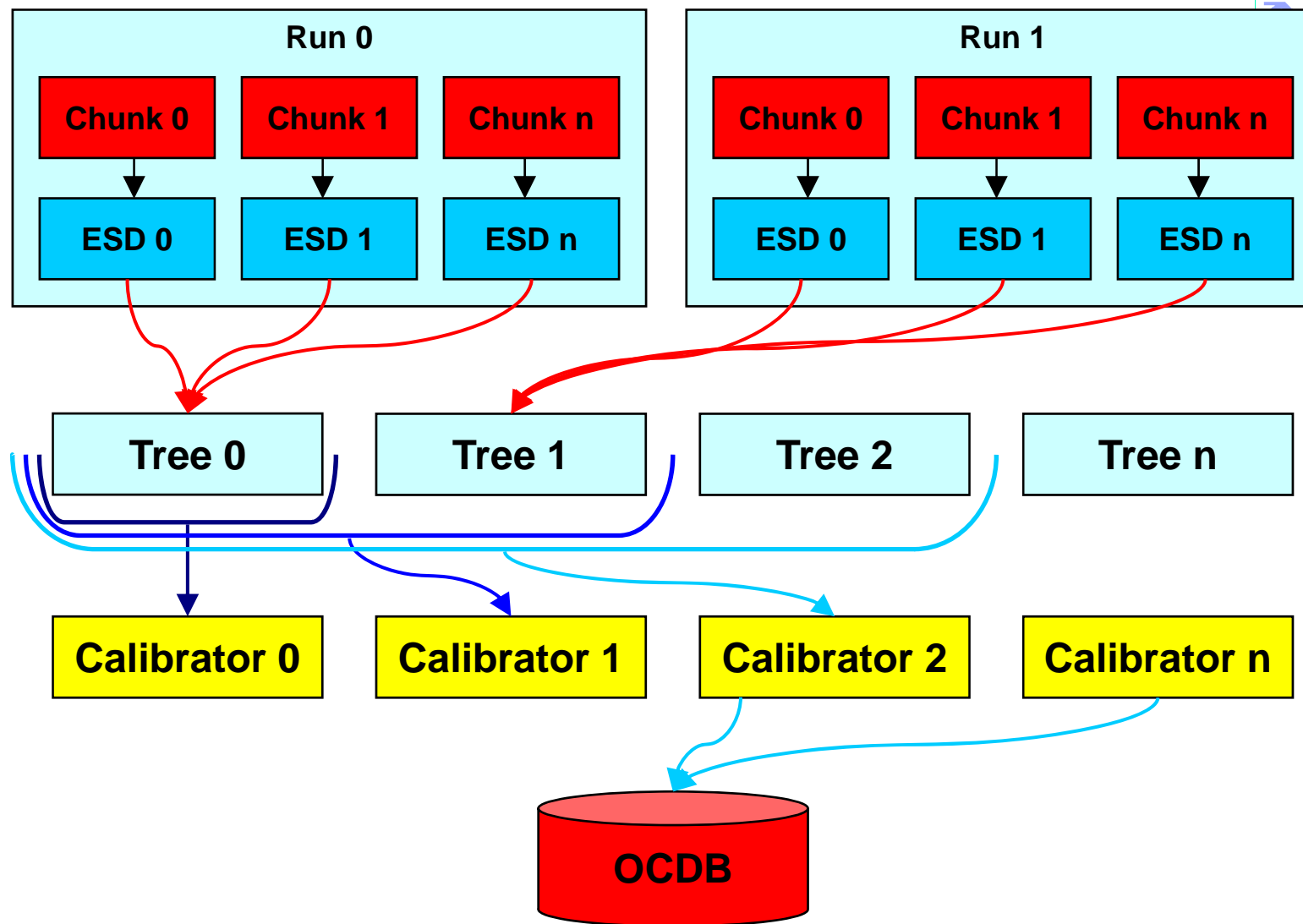
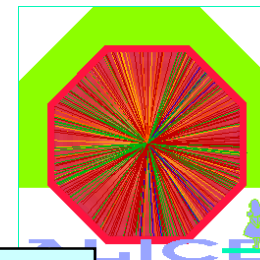
• **Process**, loop over events:

- Applying quality cuts, and retrieving essential information from ESD tracks: ToT, measured time, expected times (π , K, p) → filling of a **2D array** [TOF channels][# expected tracks/channel per run], *as before*;

• **Terminate**:

- Perform **Combinatorial PID** → reducing the 2D array size of a 1.7 factor (only one exp time stored), *as before* but added some improvements in the encoding of the algorithm;
- Filling a **run-level tree**, one entry/TOF channel, *changed*;
- Storing the tree in **AliEn**, *new*.

TOF Offline calib strategy



TOF Offline Calibration: AliTOFcalib

- ✦ **OCDB functionalities for Online/Offline Calibration:**
 - ✦ writing/reading of TOF CDB objects, both online and offline, both for decalibration (in MC) and calibration
- ✦ **Functionalities for Offline Calibration:**
 - ✦ Summing over the run-level trees stored into AliEn
 - ✦ Now, creating a **TTree** object, copying the entries of the existing ones (TTree::CopyEntries()):
 - ✦ Quite memory expensive
 - ✦ Allows to create the final tree also in case the run-level trees are stored in the reference data using the CDB classes
 - ✦ Thinking of using a **TChain** object:
 - ✦ No memory expense
 - ✦ Many Log Info messages from TAlienFile
 - ✦ Only possible in case the run-level trees are stored in AliEn without using the CDB classes (but this seems to be the way..)

TOF Offline Calibration: AliTOFcalib

- **Functionalities for Offline Calibration** (cont'd):
 - Perform Calibration (*as before*):
 - Calibration the whole TOF → default
 - Calibrating a selected channel → debug purpose
 - Calibrating summing over more than one channel → in case uniform behaviour between channels is expected
 - Calibrating a specific set of channels → partial TOF configuration
 - Calibrating using a dedicated algorithm to produce profile histogram → debug purpose
 - Still the same **debugging functionalities**:
 - possibility to store control histos both during the 2-D array filling, and during calibration

TOF Calibration Objects

- ~ Online/Offline calibration objects have been rearranged. They are **TObjArrays** of ~160000 AliTOFChannelOnline/ AliTOFChannelOffline
 - ~ **AliTOFChannelOnline:**
 - ~ Channel status (Bool_t, so far, to be changed into char)
 - ~ Time delay calculated by the TOF Preprocessor
 - ~ **AliTOFChannelOffline:**
 - ~ 6 Time Slewing correction parameters (5th order polynomial fit)
- ~ The objects are stored in **TOF/Calib/OnlineDelays** (online objs) and in **TOF/Calib/ParOffline** (offline obj):
3-level structure.

SAME DATA SIZE IN OCDB AS BEFORE

TOF Calibration Objects Validity

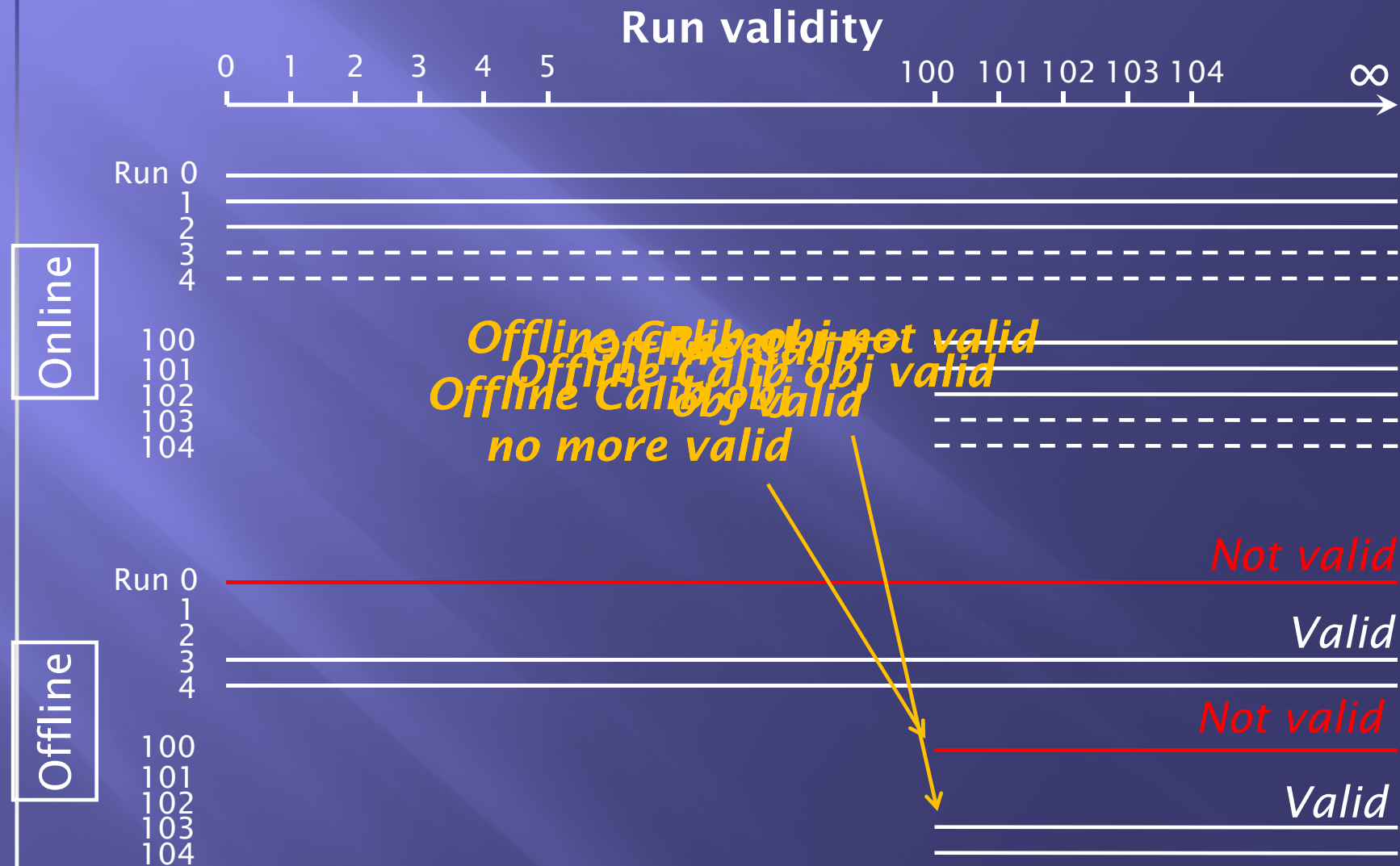
~ Run Range:

- ~ Both online and offline calibration objects have by default *infinite validity*, since they have to be used for subsequent runs;
- ~ Nevertheless, possibility to set First and Last Validity Run, especially to be able to handle correctly the switch between calibration with online/offline objs (see later).

~ Switch Online/Offline Calibration:

- ~ A “dummy” (“invalid”) offline calibration object is to be stored in the OCDB by the TOF experts at the very beginning;
- ~ Online calib objs used till a “valid” offline calib obj is put in the OCDB; at that moment, the offline calibration takes over;
- ~ If, for some reason, a *reset* of the offline calibration has to occur, a new “invalid” offline calibration object has to be put in the OCDB, and online calibration is restored; both for online/offline objs the run range changes, the first run becoming the current one.

Managing the OCDB during reco



Offline Calibration Obj: fValidity

- Handled by AliTOFcalib as a const char* member, fValidity;
- Validity for offline calibration objects is written as a comment (AliCDBMetaData::SetComment()) *in the MetaData* of the obj;
- *“valid”* is the key-word to activate the use of offline calibration params during reco;
- fValidity set as “valid” when AliTOFcalib::Calibrate ends successfully (i.e. enough statistics), otherwise no new offline calibration objs stored;
- During TOF reconstruction, *both online/offline* calibration objects are *read*: the first being always necessary to retrieve the status of the channel, the second being used only in case fValidity==“valid”.

TOF Alignment

- ~ **Tracking2Local** transformations have been defined in AliTOF::AddAlignableEntries(), committed in CVS since ~1 week. New inheritance of AliTOFcluster from AliCluster (more details in Silvia's presentation this afternoon).
- ~ In AliTOFAlignment, treatment of survey data has been improved incorporating the access to real **Survey Data** via AliSurvey classes. Introduced some functionalities (combine measurements from different files, ...).
- ~ Now start working on alignment procedures **with tracks** (Silvia + a graduating student).

TOF Code Clean-Up

- ~ The rearrangement of the TOF Offline Calibration has triggered a rearrangement of the TOF Code:
 - ~ Some **classes** removed from compilation, e.g:
 - ~ AliTOFChannel: now tho different calibration objects AliTOFChannelOnline/AliTOFChannelOffline;
 - ~ AliTOFCal (and related): now a TObjArray is being used;
 - ~ **Calibrate functionalities** moved from AliTOFCalibTask to AliTOFcalib;
 - ~ TOF **local Reconstruction** changed according to new Calibration schema, checking the Offline calib obj validity;

TOF Code Cleaning-Up (cont'd)

- ~ Taking advantage of this “revolutionary” moment, some changes in the AliTOFGeometry class have been done:
 - ~ AliTOFGeometry and AliTOFGeometryV5 have been merged: new AliTOFGeometry class.
 - ~ Some members have been turned into static (simpler code)

- 19 classes affected
- TOF DA
- 8 macros committed



Conclusions

- ~ The TOF Online/Offline Calibration schema is going to be finalized – some more refinements may be included, but the basic ingredients are there;
- ~ Some more efforts has to be done in conjunction with DAQ/DCS people for what concerns type 3 procedure for the TOF online calibration;
- ~ TOF Alignment: plan to work on alignment with tracks;

*A feeble light towards
the end of the tunnel...*

