# ALICE: alignment and computing

Raffaele Grosso

ALICE Offline Group

$2^{nd}$ LHC Detector Alignment Workshop
June $26^{th}$ 2007

# Introduction

Alignment information is produced at several steps, is interlaced with calibration and tracking. Our tools and plans to create and apply alignment information coherently:

- main features of the Alice Offline Alignment Framework;
- main implementation features of the ALICE Offline Conditions Data Base;
- framework for alignment procedures;
- sources of alignment information;
- alignment data-flow.

# ALICE offline alignment framework

- Manages production of alignment objects (from survey data, alignment procudures), storage to/retrieval from OCDB, application to geometry and querying geometry for alignment information
- interfaces to the geometry and the OCDB The framework has been updated with respect to:

- improved access to the geometry for alignable volumes;
- the alignment objects (how alignment conditions are stored);
- application of alignment objects (synchronization with the geometry, nested volumes)

# Geometry - Alignment conditions

## Geometry

- ALICE geometry is written using the ROOT geometry package (hierachical geometry independent from transport);
- alignment information can be read and applied from ROOT objects to the geometry, then coherently used in simulation and reconstruction.

## Alignment objects

Hold alignment conditions:

- volume identifier (unique volume index + symbolic volume name)
- global delta transformation

Symbolic volume names allow the objects not to depend on changes in the geometry tree.

Global transformation can be saved both as a matrix or a minimal set of parameters (6 doubles).

# Fast access to geometry information

- mapping indices - ”alignable entries”
- caching of ideal global position

- initialized once when the geometry is loaded;
- allow direct access to physical nodes for alignable volumes.

Alignable entries

- link the symbolic volume name with the volume path
- hold pointer to physical node
- hold the tracking-to-local matrix

# Storage and retrieval to/from OCDB

Our OCDB is a set of read only ROOT files residing on the grid:

- takes advantage of database and metadata capability of the file catalogue;
- is accessible from the TIER 1 and 2 centers without need of duplication and synchronization;
- versioning is handled authomatically by the framework;
- synchronization is based on
  - run-range and version, both included in the filename;
  - metadata.

The framework defines a default transparent behaviour; the user can easily modify it on a sub-detector basis.

# Zero-pass alignment

Alignment will happen in several steps:

## Zero-pass alignment

Several types of geometry corrections will be produced before reconstruction of collision events is available:

1. information from survey and monitoring systems (MUON, ITS and space frame) - see talks by Raphael and Ricardo;
2. visual inspection of raw data (event display) - possibility of human intervention;
3. information from cosmics and alignment runs.

# Zero-pass alignment

Feasibility and interest of on-line procedures is still under evaluation. Presently we foresee:

- to run a procedure based on the geometrical overlaps of neighbor sensors in the ITS.
- possibility to run a "fast tracking" on-line on the GDCs or on the monitoring farm. Fast cluster association could be run on:
  - TPC seeds
  - HLT tracks as seeds

  using cluster search windows enlarged according to survey and calibration precision.

On-line procedures will not be run with the first data but possibly at a later stage.

This implies that first-pass reconstruction will mainly relay on good knowledge of geometry from survey and cosmic+beam tracks.

ALICE

# First-pass alignment

## First-pass alignment

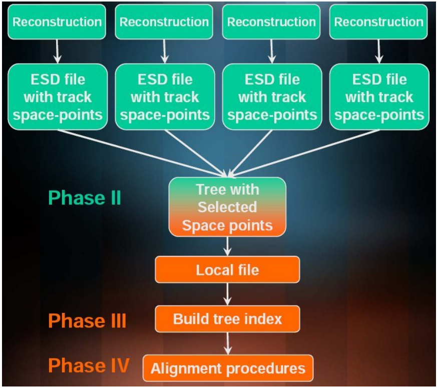
First-pass reconstruction is run using geometry corrected by zero-pass alignment. Resulting tracks and associated space-points are used by track-based alignment algorithms:

- common framework for internal alignment of barrel detectors and for global alignment;
- parallel alignment of the inner tracking system using Millepede algorithm;
- alignment of the MUON chambers using Millepede.

# Framework for alignment procedures

The framework for alignment procedures provides the tools for the whole process from the ESD files to the production of the alignment objects, managed by a single steering class:

- space-points extraction and processing;
- track fitting;
- track extrapolation points;
- residuals minimization.



# Framework for alignment procedures

The alignment strategy is sub-detector and data dependent. (see Andrea's talk).
It is implemented in macros calling methods of the steering class in order to set:

- the order in which volumes are aligned;
- the track-fitting algorithm;
- the number of iterations;
- the set of volumes to be aligned and the set of reference volumes + constraints.

ALICE

# Retrieval of space points

Space points contain:

1. the global coordinates;
2. the covariance matrix (6 floats);
3. the unique volume identifier.

and are associated to a track.
Reasons for the choice of the global RS:

- same RS for all the barrel (e.g. all track points);
- fast track fitting;
- local RS is not guaranteed to be unique.

The local RS can be extracted from this information.

# Retrieval of space points

The framework manages:

1. retrieval,

of the space points.

1. The space points are stored inside the ESD track during reconstruction if a flag for the reconstruction job is activated (for alignment runs).

# Retrieval of space points

The framework manages:

1. retrieval,
2. filtering and

of the space points.

1. The space points are stored inside the ESD track during reconstruction if a flag for the reconstruction job is activated (for alignment runs).

2. Tracks are filtered and space points are saved locally in a tree of arrays of spacepoints, corresponding each to a track.

ALICE

# Retrieval of space points

The framework manages:

1. retrieval,
2. filtering and
3. indexing

of the space points.

1. The space points are stored inside the ESD track during reconstruction if a flag for the reconstruction job is activated (for alignment runs).
2. Tracks are filtered and space points are saved locally in a tree of arrays of spacepoints, corresponding each to a track.
3. An index is built which associates to each volume all the array of points corresponding to a track passing through the given volume. This allows the alignment procedure to load only space points of interest for the volume which is being aligned.

ALICE

# Track-fitting and residuals minimization

Several fitters and minimizers available; they can be choosen via the steering class. Available fitters:

- linear fitter in conformal mapping space (fast & robust);
- straight track fitter (for B off);
- Kalman filter track fitter (for cosmics).

Availble residual minimization methods:

- minuit based;
- linear minimization (robust option);
- fast linear minimization.

ALICE

# Data-flow

1. for pp runs a ”quasi-online” reconstruction (i.e. reconstruction jobs will start in continous mode as soon as raw data are stored) will initially use the alignment objects produced by zero-pass alignment;
2. when the number of events needed by alignment procedures has been reached, a predefined set of alignment algorithms will be launched, either on the grid or on the CAF (500 CPUs, 100 TB), and will produce alignment objects which will be included in the geometry for the remaining reconstrucion jobs.

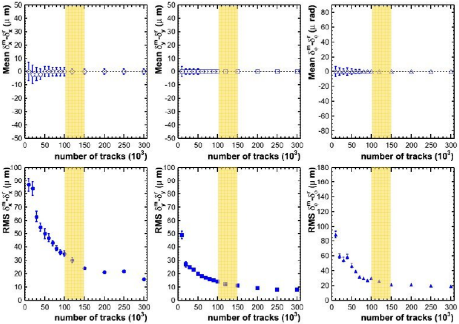
ALICE

# Data-flow

- This will happen at the beginning of every LHC period and each time a significant misalignment of the geometry is considered to be possible (when switching off/on the magnetic field, after installation of new components).
- A predefined set of alignment algorithms will be launched between first and second pass reconstruction.

# Schedule of alignment runs

- Estimate based on MUON chambers track-base alignment simulation (B off): alignment achieves required resolution (30$\mu$m in y) with $\sim$20 thousands tracks
- We expect to need a dedicted LHC fill for alignment a few times per year. Stability tests will be run to define frequency.




For central barrel only rough estimates yet: required resolution achieved with $\sim$100-500 thousands tracks

# Conclusions

- Alignment framework and OCDB stable, tested during PDCs
- Framework for alignment procedures developed and tested
- Alignment procedures using:
  - common framework for internal alignment of barrel detectors and for global alignment;
  - Millepede algorithm for parallel alignment of the inner tracking system;
  - Millepede for alignment of the MUON chambers.

  Studies started.
- Number of tracks required for needed resolution roughly estimated
- Alignment data flow planned

Thanks to all ALIgners