USING THE RECONSTRUCTION SOFTWARE, ORCA, IN THE CMS DATA CHALLENGE

St. Wynhoff, Princeton University, (CERN, Geneva, Switzerland)

Abstract

We report on the software for Object-oriented Reconstruction for CMS Analysis, ORCA. It is based on the Coherent Object-oriented Base for Reconstruction, Analysis and simulation (COBRA) and used for digitization and reconstruction of simulated Monte-Carlo events as well as testbeam data.

For the 2004 data challenge the functionality of the software has been extended to store collections of reconstructed objects (DST) as well as the previously storable quantities (Digis) in multiple, parallel streams.

We describe the structure of the DST, the way to ensure and store the configuration of reconstruction algorithms that fill the collections of reconstructed objects as well as the relations between them. Also the handling of multiple streams to store parts of selected events is discussed. The experience from the implementation used early 2004 and the modifications for future optimization of reconstruction and analysis are presented.

THE ORCA PROJECT

The ORCA project [1] in the CMS experiment [2] is focussed on the reconstruction and physics analysis of the events that will be recorded with the CMS detector once the LHC accelerator will be running. It also takes care of the combination of signal and pileup Monte-Carlo events and the simulation of the detector readout and the Level-1 trigger electronics.

The main parts of ORCA however, focus on the reconstruction of detector objects, e.g. Tracks, Clusters, Muons, Vertices, missing energy, the algorithmic code for use in the HighLevelTrigger and on "offline" reconstruction of physics objects like Jets, Electrons or Photons.

To achieve this optimally, the SEAL, POOL and PI LCG [3] software projects are used – mostly services to store and retrieve the generated information. On the CMS software side ORCA depends on the COBRA package which provides the basic services and interfaces and Geometry where the detector geometry is described in XML. The event simulation is done with the OSCAR project – using the same base projects as ORCA – and storing the simulated events in a ORCA readable format. Finally, to visualize the events and the CMS detector in all formats the IGUANACMS project is used [4].

Of the various CMS projects ORCA has the highest and most stringent requirements. About 200 developers provided until now roughly 6000 files in 375 packages

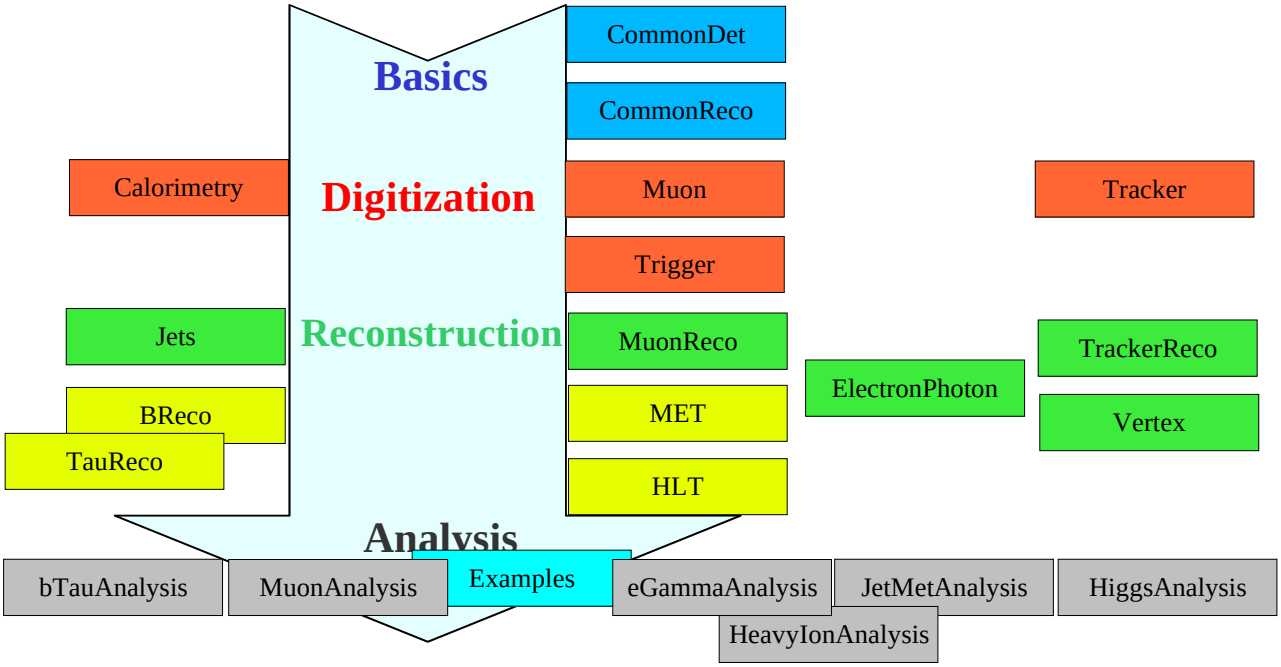

Figure 1: ORCA project structure

(i.e. libraries). The project is grouped into 22 subsystems – i.e. collections of libraries – that focus on the various subtasks of the project or are dedicated to physics analysis (see Fig. 1).

FROM SIMULATION TO RECONSTRUCTION

To allow the physics studies to prepare for the LHC startup, Monte-Carlo events are generated. This is still FORTRAN based and the events (signal as well as minimum bias) are stored in Ntuples. These event samples are then treated with the OSCAR project. It simulates the reaction of the CMS detector to these events with GEANT4 [5] and stores the result using the POOL tool in CMS-specific files. These files contain the original Monte-Carlo information and the information generated by the simulation, i.e. Tracks and vertices as determined by GEANT4 and the simulated reaction by the various subdetectors: hits. About 200 thousand minimum bias events and 90 million signal events have been simulated until now.

The next step is the simulation of the response of the detector on electronics for a particular luminosity. This step is called digitization and combines the minimum bias events and the signal events depending on the luminosity. This is done with the *writeAllDigis* program of ORCA. The results – Digis and Associations between the Digis and the simulation input – are again stored in POOL files. This step is done for about 80 million signal events until now. The samples of the digitized events are readable as input of the reconstruction jobs. Since several years analysis was done reading the files containing digitized event samples.

DST DEVELOPMENT

Doing the physics analysis on the produced Digi samples has advantages but also disadvantages. Since running the reconstruction code (to reconstruct Tracks, Jets, etc.) takes a significant amount of CPU time it is a big advantage to store the reconstructed objects for fast access and analysis. One important target for the data challenge 2004 was to store as much reconstructed information in a new format called DST (data summary tape) as possible. The first version to be used was developed in about 2 months. These DSTs contained the possibility to reconstruct and store in particular Tracker tracks (CombinatorialTrackFinder) - very important since this algorithm is the most CPU intensive one. In total about 40 different kinds of objects were made storable.

The time requirements for the different objects to be reconstructed (measured on Xeon 3.2 GHz) are shown in Fig. 2. Since the Level-1 trigger information was not

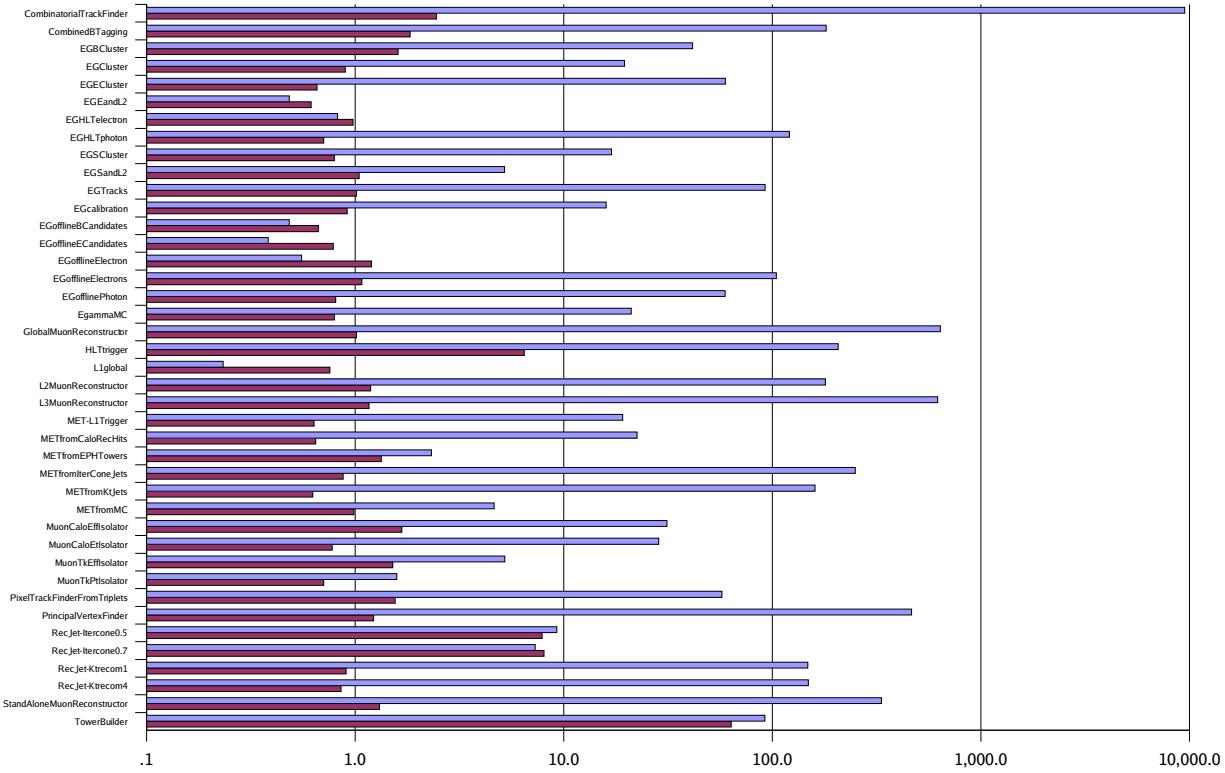

Figure 2: Time needed by the various reconstruction algorithms doing full calculation (blue) or when reading from DSTs (red). The time is given in seconds for totally 500 events H -> ZZ -> $mu^+mu^-e^+e^-$

stored in the digitized input events this step is included in the time. Two different sorts of selections are possible.

- What to calculate – for this the code has to be modified and linking adapted.
- What to store – for this a standard configuration file is used.

For particularly fast analysis the information stored in the DST can be read. Since this does not require to calculate the stored objects again a speed improvement of about a factor 100 -1000 is achieved.

In addition to the possibility to store the reconstructed objects, a special executable – *writeStreams* – is capable of storing not only the content like *writeDST* but in addition for selected events only a sub-part of all calculated information. The event-based selection can be different for the various streams . Also for each stream a different part of objects - reconstructed ones or even Digis – can be selected individually.

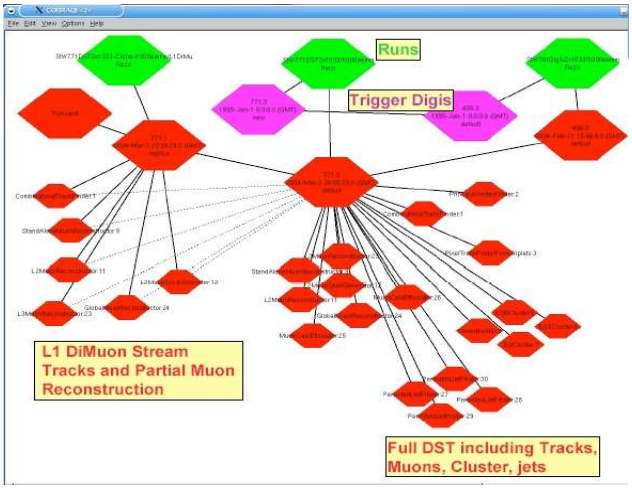

Figure 3: DST stream and its relates to the full information.

Fig. 3 shows the relations between the full DST content including Tracks, Muon, Clusters, Jets, etc. and a particular stream based on the Level-1 trigger to select events with at least two Muons. In that stream only Tracks and Muon information is stored but navigation to the full information (DSTs as well as Digis) is possible.

DST USAGE IN DATA CHALLENGE 2004

This year in March and April the CMS data challenge has been running. It was focussed on *writeStreams.* After fast debugging of the memory usage the production team started to use ORCA 8.0.1 that turned out to be running with high reliability. In total about 20 million events were reconstructed (some repeatedly). The CPU time consumption was dominated by the Tracker track reconstruction (CombinatorialTrackFinder) using about 90%. In total 8 special streams were stored: 6 matching to the Level-1 trigger information and two for calibration tasks.

The resulting files with the reconstructed information were distributed to 5 outside computing centres to allow analysis to be done there.

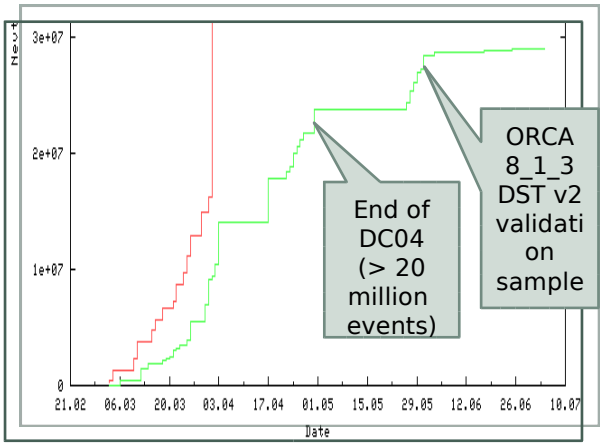

Figure 4: Number of events produced and stored in DSTs

SUMMARY

ORCA, the reconstruction software project of CMS, was up to 2003 focussed on digitization.

The implementation of DSTs allows to make reconstructed objects persistent. The main collection contains all reconstructed objects asked to be stored via datacards. In addition different subsets of objects are stored in parallel streams. Reading of DSTs allows fast analysis using the stored data.

The data challenge 2004 was the first main test where the ORCA *writeStreams* executable was running in a very stable way. In total more than 20 million events were processed successfully, the DST content and streamed information stored and distributed for test analyses.

ACKNOWLEDGEMENTS

Thanks to the CMS ORCA project developers team and the CMS production team.

REFERENCES

[1] http://cmsdoc.cern.ch/orca
[2] http://cmsdoc.cern.ch
[3] http://lcgapp.cern.ch/project
[4] http://cmsdoc.cern.ch/cmsoo/cmsoo.html
[5] http://geant4.web.cern.ch/geant4/geant4.html