

Reco profiling notes

- Use 16.0.2.7.
- Run RAWtoESD.
- Use first 20 events from
`data10_7TeV.00165632.express_express.merge.RAW`
`._1b0200._SF0-ALL._0001.1_1b0025._0001.1`

CPU profiling

Run over 20 events; Top 10 algorithms, + other calo/egamma algs:

```
1 6076 (16.13) Trk::TrkAmbiguitySolver::execute() [52]
2 4684 (12.43) AthenaOutputStream::execute() [78]
3 4145 (11.00) JetAlgorithm::execute() [90]
4 2448 ( 6.50) AthenaMonManager::execute() [103]
5 2125 ( 5.64) InDet::SiSPSeededTrackFinder::execute() [112]
6 1328 ( 3.53) InDet::TRT_TrackSegmentsFinder::execute() [168]
7 1036 ( 2.75) TrigBSExtraction::execute() [199]
8 1028 ( 2.73) CaloClusterMaker::execute() [202]
9  928 ( 2.46) CaloCellMaker::execute() [212]
10  910 ( 2.42) InDet::InDetExtensionProcessor::execute() [222]
12  814 ( 2.16) LArRawChannelBuilderDriver::execute() [239]
19  474 ( 1.26) egammaBuilder::execute() [355]
22  425 ( 1.13) softeBuilder::execute() [395]
26  287 ( 0.76) TileRawChannelMaker::execute() [575]
27  277 ( 0.74) CaloTowerAlgorithm::execute() [586]
31  226 ( 0.60) CaloTopoTowerAlgorithm::execute() [691]
```

CPU profile notes

AthenaOutputStream: About 35% running gzip, 21% loading converters (one-time operation), 12% in `ClassIDSvc::getTypeNameOfID`. (Mostly `std::map` operations.)

Balance is mostly TP converters. Top 5 types:

```
159 (11.52) Track
125 ( 9.06) CaloCellContainer
100 ( 7.25) TrackParticleContainer
 86 ( 6.23) DataHeaderCnv
 84 ( 6.09) CaloCellLinkContainer
```

Other notes (same as from July PMB meeting):

- CaloCellContainer dominated by packing the cells.
- TrackParticleContainer dominated by calls to `pool::RootTreeContainer::size` (which is mostly calling `GetBranch`). Seems kind of funny: not clear why this behaves differently than other containers.
- DataHeaderCnv has been improved somewhat, but still seems to be dominated by memory allocation for string operations.

CPU profile notes

AthenaMonManager: Top 5 fillers:

```
652 (25.37) LArCoverage::fillHistograms() [288]
645 (25.10) IHLTMonTool::fillHistograms() [291]
127 ( 4.94) CalorimeterL1CaloMon::fillHistograms() [1050]
113 ( 4.40) SCTHitEffMonTool::fillHistograms() [1134]
110 ( 4.28) TRT_Monitoring_Tool::fillHistograms() [1151]
```

- LArCoverage dominated by LW histogram filling and by call to getNoise.
- CalorimeterL1CaloMonTool spends time in root histogram filling and in is_tile. is_tile is the first DDE method called in a cell loop — probably from loading DDE into cache.

CPU profile notes

CaloClusterMaker:

- Mostly topo cluster making and CaloClusterLocalCalib.
- Topo clustering dominated by getEffectiveSigma from the noise tool and CaloCluster::updateKine.
- Observe that get_element_FCAL is slow. (Searches all FCAL cells! But only a small fraction of the calorimeter.)

CaloCellMaker

- About half in CaloCellContainerCorrectorTool. Most of this in CaloCellPedestalCorr and LArCellHVCorr; most apparently seems to be fetching data from DB objects.
- Next is CaloCellContainerFinalizeTool, sorting the CaloCellContainer and building the lookup table. Prefetching iterator could help here?

CPU profile notes

LArRawChannelBuilderDriver:

- Mostly applying OFCs.

egammaBuilder

- Brem fit: 24%
- Calo iso: 27%
- egamma iso tool: 9%
- Track refit: 7%
- Conversion refit: 13%
- Track isolation: 5%
- Calo isolation makes many calls to `CaloCellList::select`. Could they be combined?

softeBuilder similar to egammaBuilder.

CPU profile summary

- Can we speed up LW hist filling?
- Should `CalorimeterL1CaloMonTool` use LW histograms?
- Some loops over cells might benefit from prefetch iterator; eg, `CalorimeterL1CaloMonTool` and `initializeLookUpTable`.
- What's going on with `TrackParticleContainer` writing?
- Can noise constant access be sped up?
- Rewrite `get_element_FCAL`.

Memory allocation by Athena component before first event

IOVDbSvc	72M
InDet::TRT_LayerBuilder	66M
PoolSvc	46M
JobOptionsSvc	23M
CaloMgrDetDescrCnv	23M
JetBadChanCorrTool	22M
MboySvc	21M
AthenaPoolConverter	21M
Muon::MuonStationBuilder	19M
LArHVTToolDB	19M
<hr/>	
LArADC2MeVTool	12M
LArRoI_Map	9M
CaloAlignTool	7.7M
CaloNoiseToolDB	4.5M
egammaEPiBoostedDecisionTreeTool	3.8M
CaloTTMgrDetDescrCnv	2.5M
LArDetectorToolNV	2.3M
LArHVCablingTool	1.9M

Memory allocation by Athena component after first event

PoolSvc	105M	+59
LArOFCSubset	103M	+103
LArShapeSubset	103M	+103
IOVDbSvc	75M	+3
InDet::TRT_LayerBuilder	66M	+0
CaloMgrDetDescrCnv	23M	+0
CaloTowerAlgorithm	23M	+23
JobOptionsSvc	23M	+0
JetBadChanCorrTool	22M	+0
AthenaPoolConverter	21M	+0
<hr/>		
LArCellBuilderFromLArRawChannelTool	21M	+21
LArHVTToolDB	19M	+19
LArADC2MeVTool	12M	+0
LArRoI_Map	9M	+0
CaloAlignTool	7.7M	+0
LArRodDecoder	7.0M	+7.0
LArPedestalSubset	4.7M	+4.7
CaloNoiseToolDB	4.5M	+0

Memory allocation comments

- PoolSvc memory is the output trees.
- Lots of memory for OFC/shape tables. Could probably be reorganized to be significantly smaller. But may entail interface/DB changes. (But see next slide.)

Summary of changes since July:

- MuonCalib problem fixed.
- PoolSvc down by 8M, IOVDbSvc up by 21M.
- Seeing 23M from JobOptionsSvc that I didn't see before.

LAr OFC comments

- LAr is currently using a lot of memory for OFC coefficients and wave shapes.
- To calculate energies from the LAr time samples.
- Ultimately, this is supposed to be done entirely in the DSP code; and not in reconstruction.
- Summary from a mail from Guillaume:
 - ▶ Reprocessing uses a smaller set of conditions data for this calculation. Reduces footprint by $O(100M)$.
 - ▶ Heavy ion processing relies entirely on the DSP calculation; so these conditions data should not be loaded in that case.
 - ▶ Default reco configuration remains unchanged for now.