# Compression level of ROOT output

Gianluca Petrillo

`LArSoft` Librarians' meeting, May 6$^{th}$ , 2014

# Output files

- products are propagated by `art` from input to output
- `RootOutput` module has more and more work as we go down the production chain
- it can take relevant time

Our observations:

- Wesley has noticed that the time taken by `RootOutput` strongly depends on the compression level we request...
- ... and the compression level may have limited impact on the size of the output
- Herbert has confirmed that this kind of findings led DØ to choose a compression level 1
- `art` sets compression level to 7 by default

I have collected some figures to help characterize this behaviour.

# Test description

- tests run with `LArSoft v1_01_00` on GPVM machines
- 100 events per job
- same random seeds when appropriate
- compression level set by one line like:
  outputs.out1.compressionLevel: 7
  (`out1` being the label of the RootOutput module)

Figures have $\mathcal{O}$ (3%) statistical fluctuations.

| compr. level | prodsingle | | reco | | | |
|---|---|---|---|---|---|---|
| | | | input compr. 0 | | input compr. 7 | |
| | size [MB] | s/evt | size [MB] | s/evt | size [MB] | s/evt |
| 0 | 58.4 | 0.0086 | 26499.5 | 0.072 | 26231.9 | 0.075 |
| 1 | 32.4 | 0.026 | 667.7 | 0.521 | 640.9 | 0.515 |
| 2 | 32.4 | 0.028 | 667.7 | 0.547 | 641.0 | 0.561 |
| 7 | 31.1 | 0.050 | 650.7 | 0.801 | 623.9 | 0.779 |

prodsingle → `prodsingle_lbnefd.fcl`

reco → `standard_reco_lbnefd.fcl`

The `reco` test has beed performed with two different inputs:

1. from `prodsingle` output produced with compression level 0;
2. from `prodsingle` output produced with compression level 7.

| compr. level | prodsingle size [GB] | prodsingle s/evt | reco2D size [GB] | reco2D s/evt | reco2D peak mem [GB] |
|---|---|---|---|---|---|
| 0 | 16.054 | 0.39 | 32.42 | 0.7 | 2.38 |
| 1 | 0.581 | 1.06 | 26.91 | 16.8 | 2.49 |
| 2 | 0.551 | 1.05 | 26.93 | 18.9 | 2.53 |
| 7 | 0.368 | 2.77 | 26.66 | 41.4 | 2.47 |

prodsingle → `prodsingle_uboone.fcl`

reco2D → `standard_reco_uboone_2D_noopt.fcl`

The `reco` test has beed performed with input produced with compression level 7.

# Conclusions

- the size of the output also depends (mildly) on the compression of the input
- the mileage varies a lot with the job
- going from compression level 1 to 7:
  - $\Rightarrow$ produces 5–50% larger files
  - $\Rightarrow$ takes 35–85% less time
- small differences between compression levels 1 and 2
- memory impact seems to be minor

Further tests:

- how does “fast cloning” affect the size of “reco” jobs?

# Additional material

# Producers in the test configurations

`prodsingle_lbnefd.fcl` SingleGen, LArG4, SimWireLBNE35t, RandomNumberSaver, TriggerResultInserter

`standard_reco_lbnefd.fcl` CalWireLBNE35t, GausHitFinder, HitCheater, TriggerResultInserter

`prodsingle_uboone.fcl` SingleGen, LArG4, SimWireMicroBooNE, TriggerResultInserter

`standard_reco_uboone_2D_noopt.fcl` RandomNumberSaver, CalWireMicroBooNE, GausHitFinder, RFFHitFinder, fuzzyCluster, ClusterCrawler, LArPandora, TriggerResultInserter