Truth Tracking Validation

Carl Gwilliam

UNIVERSITY OF LIVERPOOL

Marie-Helene Genest

LPSC Grenoble

24th October 2013
ISF Workshop

Introduction

- Two use cases for truth tracking within ISF:
  - Very fast simulation chain with FATRAS and pile-up done at evgen level
  - AFII legacy simulation with the standard pile-up mechanism

- First go at validating truth tracking in the latter context

- Will look at issues in submitting samples …

The Issues

- Several technical issues in running samples due to need for unfiltered minbias truth inputs necessitated by truth tracking
  - Need unfiltered MCEventCollection for digitization
    - Truth-slimmed pileup samples = 6.3 TB
    - Full-truth pileup samples = 12.8 TB !
  - Need to provide this pileup truth info to reconstruction
    - At least for BCIDs that ID is interested in
- Even if throw away pileup truth information during digitization, reading in the unfiltered minbias still increases digi memory
  - Can cope with this using PileUpTools
  - But MergeMcEventCollTool, which combines signal and background MCEventCollections, can't save only in-time truth
  - Future: config MergeMcEventCollTool to allow it to keep just signal plus information from a smaller range of bunch-crossings
- Running full truth in test productions previously was painful
  - Required small number of events /job and high memory queues

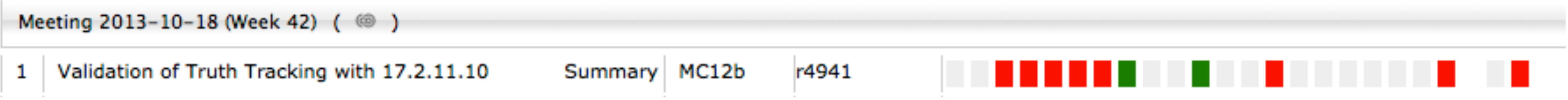
The Symptoms and Solutions

- Initially tried to run saving all pileup truth
  - Caused RDO files to exceed maximum allowed size due to large McEventCollection written
    - AthenaPoolCnvSvc WARNING FileSize > domMaxFileSize for tmpRDO.pool.root
  - Solved for validation by limiting to 200 events / job
    - Not really suitable for large-scale MC production
- Then hit memory limit from reading in unfiltered minbias in digi
  - Out of memory: saving summary ..." | "commitOutput - caught exception: std::bad_alloc
- Possible solutions:
  - In-time pile up only
    - pileupInitialBunch = pileupFinalBunch = 0
  - Save pileup within a [-50, +50] ns window
    - pileupInitialBunch = -2; pileupFinalBunch = 2
  - More complicated options allow to keep as much in-time pileup as possible while throwing away out-of-time pileup information

Validation Results

- Truth tracking for pileup and standard setup for the signal event
  - Test : r4941 (17.2.11.10)
  - Reference : r4764 (17.2.11.1)

- Got results from most groups:



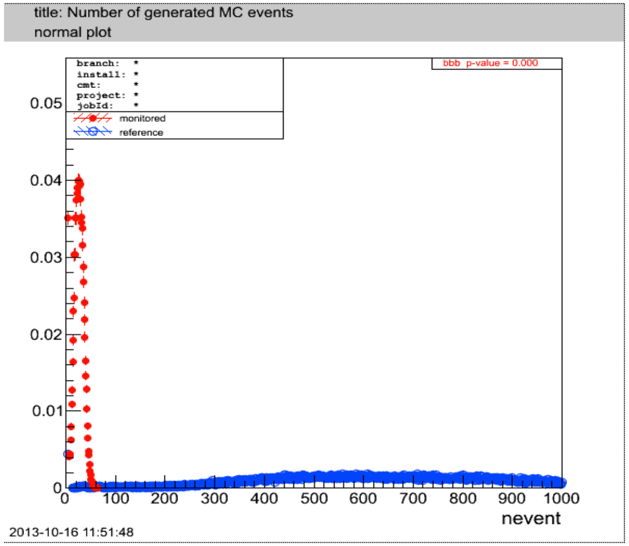
Meeting 2013-10-18 (Week 42) ( )

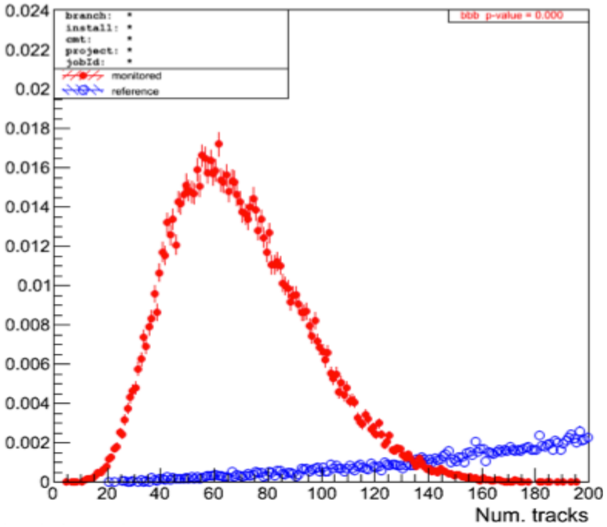
1	Validation of Truth Tracking with 17.2.11.10	Summary	MC12b	r4941	⬜⬜🟥🟥🟥🟥🟥🟩⬜⬜🟩⬜⬜🟥⬜⬜⬜⬜⬜⬜🟥 ⬜🟥

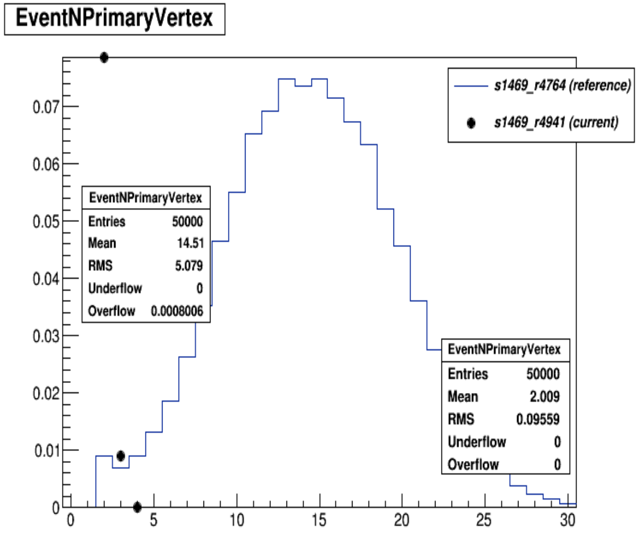
- Many differences observed (= lots of red)
  - Differences expected in the number of tracks/vertices
    - Additional effect likely in pileup, energy resolutions, efficiencies etc
  - But also saw large changes in non-tracking quantities as well (jets/MET)
  - So what was wrong …

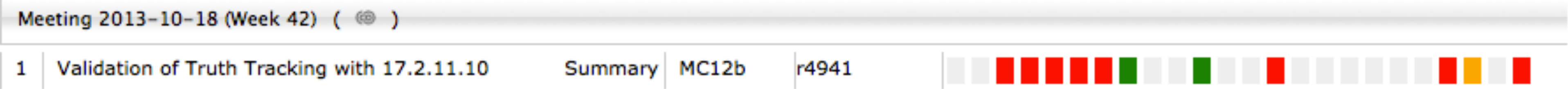
The Bug

- Unfortunatley forgot the minus sign -> no pileup!
  - pileupInitialBunch = pileupFinalBunch = 2
- Can see this in N_{gen}, N_{trk} and, most obviously, N_{PV}:









Meeting 2013-10-18 (Week 42) ( )

1	Validation of Truth Tracking with 17.2.11.10	Summary	MC12b	r4941	⬜⬜🟥🟥🟥🟥🟥🟩⬜⬜🟩⬜⬜🟥⬜⬜⬜⬜⬜⬜🟥🟧⬜🟥

- Task resubmitted for next validation meeting.
  - A week tomorrow