MDC1 and planning strategy for MDC2 tracking

Tony Frawley
Florida State University

For the tracking software group

Collaboration Meeting
June 17-18, 2021

MDC1

What did we run?

Au+Au collisions

- Central with 50 KHz pileup
- MB with 50 KHz pileup

p+p collisions

- Single Pythia events with no pileup

Reconstruction:

Truth vertexing

Otherwise, no truth information used

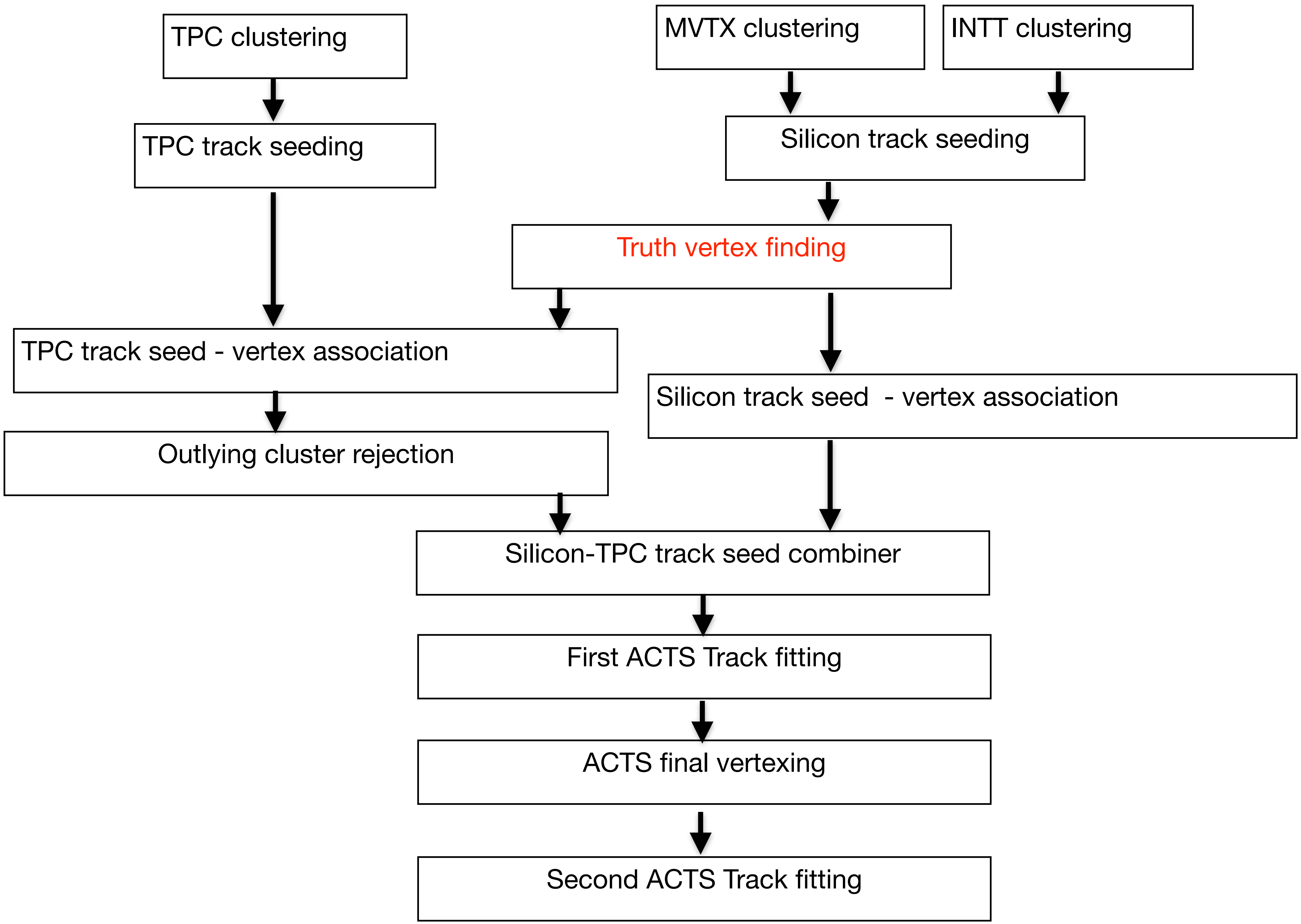
Comments:

Tracking code was not optimized for secondary tracks.

- Not a problem for Au+Au - Hijing events only
- Not a problem for p+p - Pythia assigns truth vertex for every track!
- But this needs to be fixed for MDC2

Perfectly aligned and calibrated detector, no space charge distortions

MDC1 tracking workflow



MDC2

We will be running Mock Data Challenge 2 around December of 2021.
Starting to plan for the tracking part of this now.

- What capabilities do we intend to exercise in MDC2?
- What do analyzers hope to get out of it?
- How will we evaluate the physics performance?

My opinion:

- We should aim to produce reconstructed data files that look like data
  - No truth information used at any stage in reconstruction
- All tracking steps should be in place, including calibrations
- Analyzers should aim to do physics analysis with no truth information
  - Treat MDC2 output like real data
- Assessing performance should come later
  - Probably not on the entire MDC2 data set

Capabilities for MDC2 - calibrations

All simulations done with

- Full event pileup for Au+Au and p+p
- De-aligned and de-calibrated detectors
- Space charge distortions present
- CM laser flash hits

All calibrations derived from simulated data

- Detector alignment
- Gain calibrations
- Space charge corrections - track based (average) & CM flash (fluctuations)

Calibrations applied during reconstruction

- Detector alignment
- Gain calibrations
- Space charge corrections (static + average + fluctuations)

Alignment and space charge distortions have to be disentangled
Alignment done with very low occupancy data (right?)

- Cosmics + Millipede?
- Low luminosity calibration runs + Millipede?

Calibration modules

Determine alignment corrections - how and from what?
Apply pre-determined alignment corrections - how, and to what?

- At the cluster level?

Gain calibrations

- Extract gains for each channel from data, make a map
- Apply gain calibrations to raw data as it is read in for reconstruction
- Modify hits permanently to save memory (or what?)

Space charge calibrations from tracks and laser data

- Determined after pass 1
- Applied in pass 2
- Currently in the works, see talk by Ross

Secondary tracks

Currently, all tracks are given an origin at the vertex nearest to the Z projection of the track

- Acts needs a very good starting estimate of the track origin
- This works well for primary tracks
- Not correct for secondary tracks that have an origin far removed from the nearest vertex

We need to find secondary tracks - no longer using truth vertexing

- Some may already be included with the successful Acts fits
  - Not clear how close is close enough
- Some will **not** be successfully fitted by Acts
  - Need to understand how to recover these

Possibilities:

- Rerun the seeders on all clusters not assigned to a fitted track
- Retain all track seeds and revisit the origin for the ones that failed
- Change the seed-vertex association strategy from the start

Changes to reconstruction work flow

Tracking pass 1

Begin with raw hits
Apply alignment corrections and gain calibrations
Cluster hits (identify & separate CM flash clusters)
Apply **static** distortion corrections to TPC clusters
Seed silicon and TPC tracks
Segregate silicon seeds not pointing at event vertex
Remove duplicate track seeds in silicon and TPC
Initial vertex finding with silicon seeds
Run pattern recognition, identify complete tracks
Fit silicon + micromegas clusters only in pass 1
Calculate TPC residuals and write them out
Write tracks and clusters to pass 1 output DST

CM based SCD extraction

Read CM flash clusters
Calculate distortion corrections
Write to map

Track based SCD extraction

Read Pass 1 TPC residuals
Calculate distortion corrections
Write to map

Tracking pass 2

Read tracks and clusters from pass 1 output DST
Apply **dynamic** distortion corrections to clusters
Fit tracks
Remove duplicate tracks
Calculate final vertices
Fit tracks again
(Write pass 2 output DST)

Tracking pass 3 (secondary tracks)

(Read Pass 2 output DST)
Process secondary silicon seeds
Assign best-estimate origin to track
Fit new tracks
Remove duplicate tracks
Write complete DST

Analyzer needs

MDC2 output should be an opportunity to analyze something that looks like real reconstructed data.

We need to discuss with the topical groups what the analyzers think they will be able to do with tracks

- Compare with what the tracking experts think analyzers will be able to do with tracks …

Should we write TRKR_HITSET and TRKR_CLUSTERHITASSOC?

- Used only to analyze hits that contribute to a cluster
- Do analyzers need hits?

Streaming readout in p+p tests in MDC2? What is required to handle this?

Evaluation

The simulation part of MDC2 will have to produce

- Simulated g4 hits
- Reconstructed simulated hits

These will be in separate files

The reconstruction part will produce only reconstructed clusters and tracks

We need to be able to run evaluation on the output files (perhaps only a subset)

Can read reconstructed and truth information from separate files

- But the existing evaluator is very slow for full pileup events
- Do we need an evaluator optimized for speed?
- What must it provide to satisfy analyzers?
- Experts will still use full evaluation

Tracking CPU timing (Hardware test pass 1)

Pass 1 (average time per event (s) over 3,125 jobs with ~ 75 MB Au+Au Hijing events each):

(grep "per event time" /sphenix/sim/sim01/sphnxpro/MDC1/sHijing_HepMC/HWTest/log/fm_0_20/hwpass1/condor-0000000001-0*.out)

MvtxClusterizer	0.066	
InttClusterizer	0.010	
MakeActsGeometry	0.001	
TpcClusterizer	0.867	
TpcSpaceChargeCorrection	0.157	
TpcClusterCleaner	0.076	
PHTpcClusterMover	0.008	
PHCASeeding	1.544	// will increase with better seeding efficiency
PHTpcTrackSeedVertexAssoc	0.007	
PHSiliconTpcTrackMatching	0.098	
PHMicromegasTpcTrackMatching	0.004	
PHTpcResiduals	0.039	
PHActsSiliconSeeding	1.025	
PHActsInitialVertexFinder	0.021	
PHActsFirstTrkFitter	0.208	// silicon + micromegas clusters only

Total clustering time: 0.94 s
Total tracking time: 3.19 s

Total time: 4.1 s

Tracking CPU timing (Hardware test pass 1)

Pass 1 (average time per event (s) over 3,125 jobs with ~ 75 MB Au+Au Hijing events each):

(grep "per event time" /sphenix/sim/sim01/sphnxpro/MDC1/sHijing_HepMC/HWTest/log/fm_0_20/hwpass1/condor-0000000001-0*.out)

MvtxClusterizer	0.066	
InttClusterizer	0.010	
MakeActsGeometry	0.001	
TpcClusterizer	0.867	
TpcSpaceChargeCorrection	0.157	
TpcClusterCleaner	0.076	
PHTpcClusterMover	0.008	
PHCASeeding	1.544	// will increase with better seeding efficiency
PHTpcTrackSeedVertexAssoc	0.007	
PHSiliconTpcTrackMatching	0.098	
PHMicromegasTpcTrackMatching	0.004	
PHTpcResiduals	0.039	
PHActsSiliconSeeding	1.025	
PHActsInitialVertexFinder	0.021	
PHActsFirstTrkFitter	0.208	// silicon + micromegas clusters only

Total clustering time: 0.94 s
Total tracking time: 3.19 s

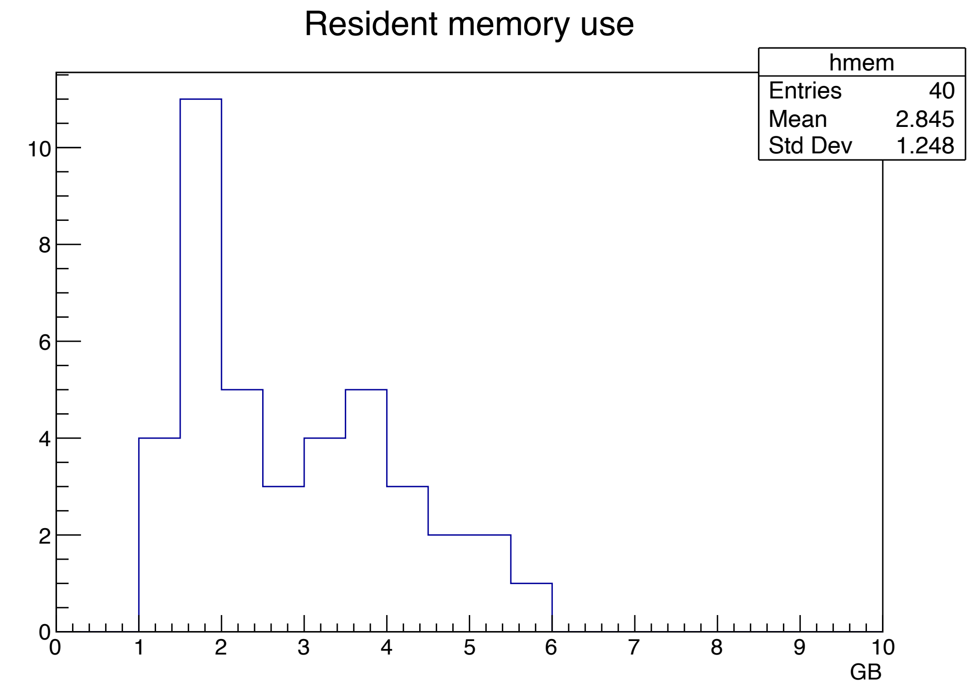
Total time: 4.1 s

The per-event wall clock time however, is in the range 10-15 s.
The difference appears to be disk I/O time.

Tracking memory use (Hardware test pass 1)

Results on tracking resident memory use that were shown at the S&C review were for single events.

Average memory per job: 2.8 GB



But when running many events per job, the resident memory grows to 9-10 GB during the first 10 events or so, and stays there.

Chris thinks this is likely due to ROOT I/O buffering. It will need to be fixed for MDC2.