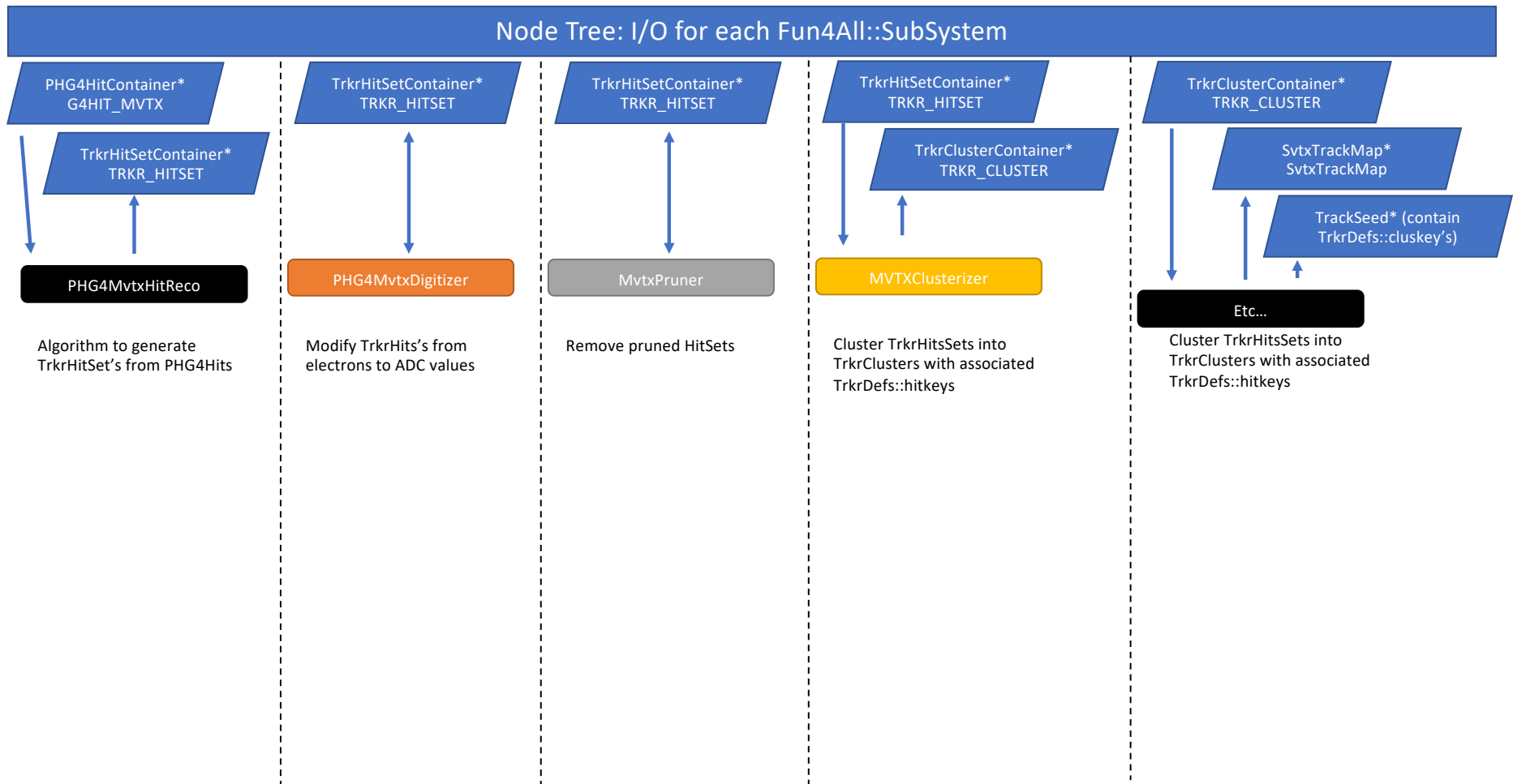


TrkrClusters for Truth Tracks

David Stewart

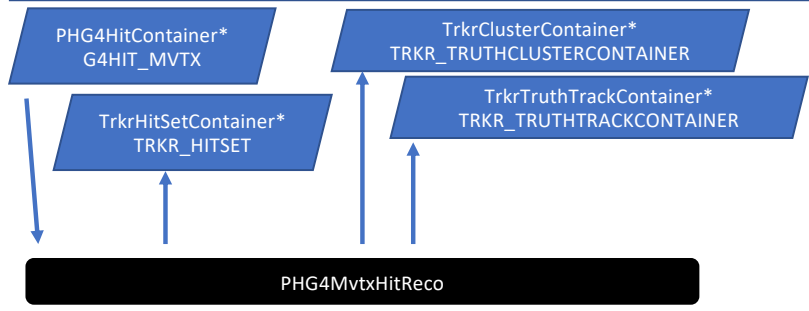
Tracking Meeting, 4/4/2023

Current process: MVTX TrkrClusters for SvtxTracks



Proposed Process: MVTX TrkrClusters for Truth Tracks

Node Tree: I/O for each Fun4All::SubSystem



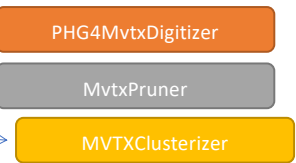
Etc...

- All the code for the SvtxTrack's (using all the TrkrHitSetcontainer for all the TrkrHits) is reimplemented in a simplified form for truth TrkrClusters.
- It is run once for each truth track's temporary TrkrHitSetContainer inside of the PHG4MvtxHitReco::process_event() loop

```

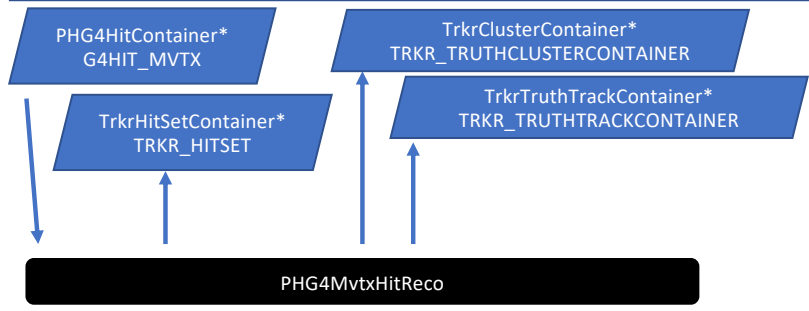
TrkrHitSetContainer all_hits = TRKR_HIT_SET; // from node tree
TrkrHitSetContainer tmp_hits = new TrkrHitSetContainer;
PHG4MvtxHitReco::process_event {
  for (PHG4Hit* g4_hit in PHG4HitContainer) {
    if (g4_hit is from a new truth track) {
      put new TrkrTruthTrack on node tree;
    }
  }
  TrkrHit* hit = process(g4_hit);
  all_hits->add(hit);
  if (g4_hit in truth track) tmp_hits->add(hit);
  if (g4_hit last g4_hit in truth track) {
    Digitize(tmp_hits);
    Prune(tmp_hits);
    Cluster(tmp_hits);
    add TrkrClusters to NodeTree;
    add clusterkeys for new TrkrClusters to NodeTree;
    tmp_hits.reset();
  }
}
    
```

Reimplement simplified version of code here.



Proposed Process: MVTX TrkrClusters for Truth Tracks

Node Tree: I/O for each Fun4All::SubSystem



Etc...

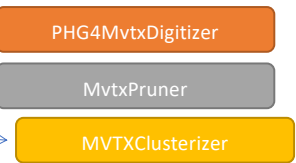
- All the code for the SvtxTrack's (using all the TrkrHitSetcontainer for all the TrkrHits) is reimplemented in a simplified form for truth TrkrClusters.
- It is run once for each truth track's temporary TrkrHitSetContainer inside of the PHG4MvtxHitReco::process_event() loop

Pros:
Don't put any TrkrHitSetContainer for truth hits on the node tree

```

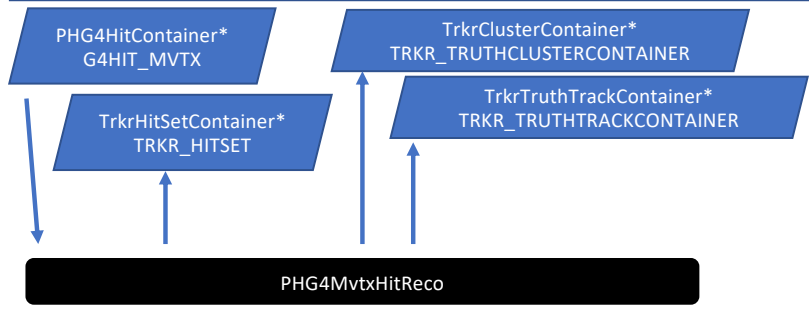
    ●
    TrkrHitSetContainer all_hits = TRKR_HIT_SET; // from node tree
    TrkrHitSetContainer tmp_hits = new TrkrHitSetContainer;
    PHG4MvtxHitReco::process_event {
    for (PHG4Hit* g4_hit in PHG4HitContainer) {
    if (g4_hit is from a new truth track) {
    put new TrkrTruthTrack on node tree;
    }
    TrkrHit* hit = process(g4_hit);
    all_hits->add(hit);
    if (g4_hit in truth track) tmp_hits->add(hit);
    if (g4_hit last g4_hit in truth track) {
    Digitize(tmp_hits);
    Prune(tmp_hits);
    Cluster(tmp_hits);
    add TrkrClusters to NodeTree;
    add clusterkeys for new TrkrClusters to NodeTree;
    tmp_hits.reset();
    }
    }
    }
  
```

Reimplement simplified version of code here.



Proposed Process: MVTX TrkrClusters for Truth Tracks

Node Tree: I/O for each Fun4All::SubSystem



Etc...

- All the code for the SvtxTrack's (using all the TrkrHitSetContainer for all the TrkrHits) is reimplemented in a simplified form for truth TrkrClusters.
- It is run once for each truth track's temporary TrkrHitSetContainer inside of the PHG4MvtxHitReco::process_event() loop

Pros:

Don't put any TrkrHitSetContainer for truth hits on the node tree

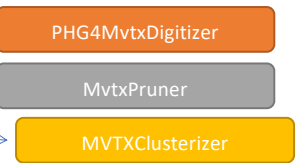
Cons:

- This is a lot of code (basically three subsystems worth) to stuff into the middle of the process_event() container. Admittedly, code can be simplified for truth clustering, but care must be taken to have accurate comparison for Svtx TrkrClusters.
- Doesn't follow Fun4All paradigm, of one process per subsystem per step, putting all I/O through the node tree.

```

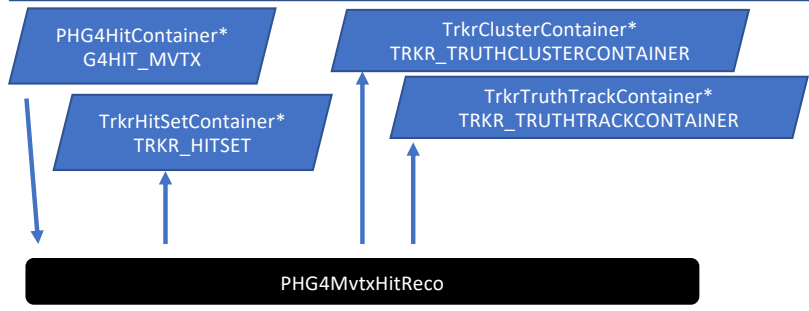
    ●
    TrkrHitSetContainer all_hits = TRKR_HIT_SET; // from node tree
    TrkrHitSetContainer tmp_hits = new TrkrHitSetContainer;
    PHG4MvtxHitReco::process_event {
    for (PHG4Hit* g4_hit in PHG4HitContainer) {
    if (g4_hit is from a new truth track) {
    put new TrkrTruthTrack on node tree;
    }
    TrkrHit* hit = process(g4_hit);
    all_hits->add(hit);
    if (g4_hit in truth track) tmp_hits->add(hit);
    if (g4_hit last g4_hit in truth track) {
    Digitize(tmp_hits);
    Prune(tmp_hits);
    Cluster(tmp_hits);
    add TrkrClusters to NodeTree;
    add clusterkeys for new TrkrClusters to NodeTree;
    tmp_hits.reset();
    }
    }
    }
  
```

Reimplement simplified version of code here.



Proposed Process: MVTX TrkrClusters for Truth Tracks

Node Tree: I/O for each Fun4All::SubSystem



Etc...

- All the code for the SvtxTrack's (using all the TrkrHitSetContainer for all the TrkrHits) is reimplemented in a simplified form for truth TrkrClusters.
- It is run once for each truth track's temporary TrkrHitSetContainer inside of the PHG4MvtxHitReco::process_event() loop

Pros:

Don't put any TrkrHitSetContainer for truth hits on the node tree

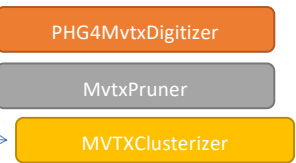
Cons:

- This is a lot of code (basically three subsystems worth) to stuff into the middle of the process_event() container. *Admittedly, code can be simplified for truth clustering, but care must be taken to have accurate comparison for Svtx TrkrClusters.*
- Doesn't follow Fun4All paradigm, of one process per subsystem per step, putting all I/O through the node tree.

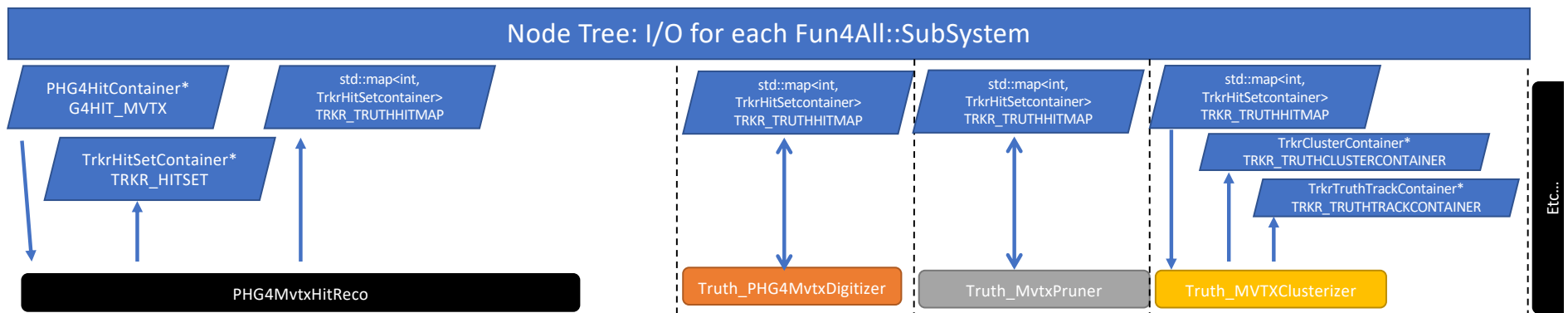
```

    ●
    TrkrHitSetContainer all_hits = TRKR_HIT_SET; // from node tree
    TrkrHitSetContainer tmp_hits = new TrkrHitSetContainer;
    PHG4MvtxHitReco::process_event {
    for (PHG4Hit* g4_hit in PHG4HitContainer) {
    if (g4_hit is from a new truth track) {
    put new TrkrTruthTrack on node tree;
    }
    TrkrHit* hit = process(g4_hit);
    all_hits->add(hit);
    if (g4_hit in truth track) tmp_hits->add(hit);
    if (g4_hit last g4_hit in truth track) {
    Digitize(tmp_hits);
    Prune(tmp_hits);
    Cluster(tmp_hits);
    add TrkrClusters to NodeTree;
    add clusterkeys for new TrkrClusters to NodeTree;
    tmp_hits.reset();
    }
    }
    }
  
```

Reimplement simplified version of code here.



Proposed Process: MVTX TrkrClusters for Truth Tracks



```

●
TrkrHitSetContainer all_hits = TRKR_HIT_SET;
map<int,TrkrHitSetContainer> truth_hits = TRKR_TRUTHHITMAP;
// also on node tree
PHG4MvtxHitReco::process_event {
    TrkrHitSetContainer tmp_hits;
    int truth_track_id;
    for (PHG4Hit* g4_hit in PHG4HitContainer) {
        if (g4_hit is from a new truth track) {
            track_id = id of new truth track;
            tmp_hits = new TrkrHitSetContainer;
        }
        TrkrHit* hit = process(g4_hit);
        all_hits->add(hit);
        if (g4_hit in truth track) tmp_hits->add(hit);
        if (g4_hit last g4_hit in truth track) {
            truth_hits.emplace(make_pair(track_id,tmp_hits));
        }
    }
}

```

Pros:

- Puts the code into the general Fun4All paradigm:
 - All the I/O happens per-event, per-subsystem, into and out of the node tree
 - Code is logically divided the same way for truth clusters as for reco clusters, even if they are really short functions.
 - Easier to update code in parallel for SvtxTrack clusters and truth clusters

Cons:

- Takes more memory by putting on the extra map<int, TrkrHitSetContainer> for a bit prior to consuming it in the following subsystems.