

Wire-Cell async node and its application on DUNE full FD Sim/SigProc

Haiwang Yu (BNL)

2024-03-25

DUNE FD Sim/Reco Meeting



@BrookhavenLab

Outline

Principals

Previous status reported at Jan collab. meeting

New Dev.:

- 2 WCT async nodes working
- configured one combined workflow for bridged wire geom
- tests with 1 numu g4.root

Discussion

Async. in WCT

WCT framework has two layers for the computing node:

- INode – engine independent
- NodeWrapper – engine independent
 - `WireCellTbb::NodeWrapper`, `Pgraph::Node`
- Previous WCT nodes need all inputs ports filled before processing
 - e.g., `IFaninNodeBase`:
`std::vector<boost::any> -> boost::any`
- New WCT Hydra node (m->n) intended to be an async. node by 2D input/output type
 - `std::vector<std::deque<boost::any>>`
 - still using null pointer as EOS (end of stream)

NodeWrapper

translation

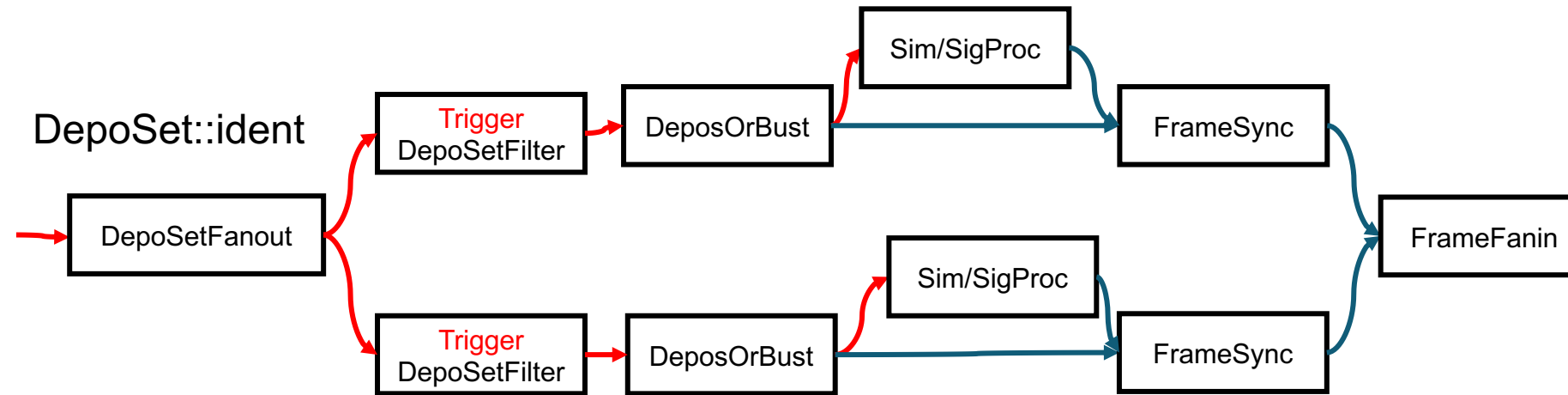
INode

translation

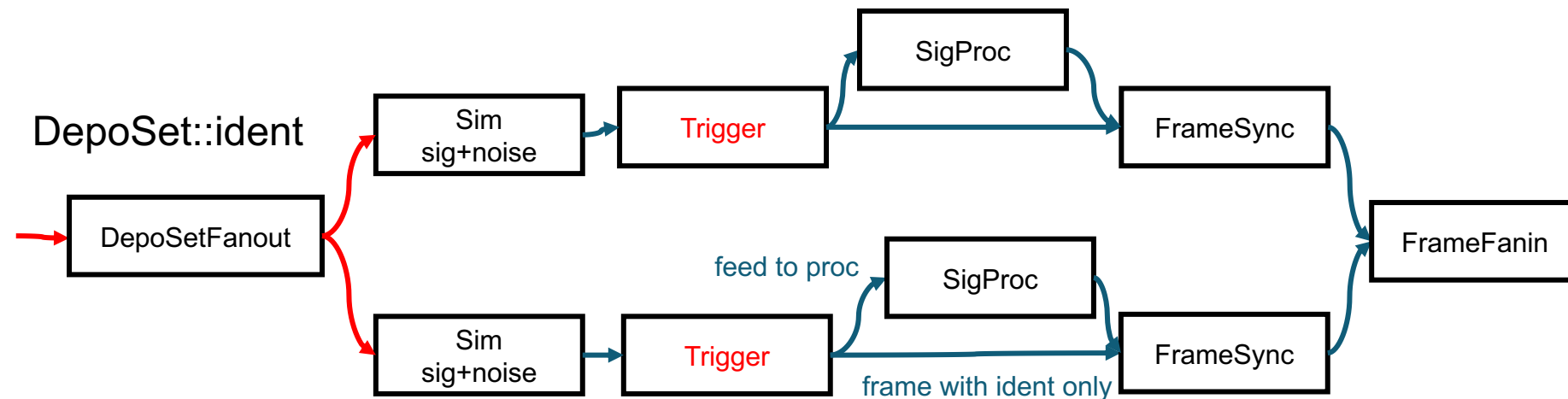
Shortcuts to skip some processing

[issue #148](#)

Cheat trigger to skip both Sim/SigProc



- Sim all APAs
- Trigger alg. to determine whether to do SigProc



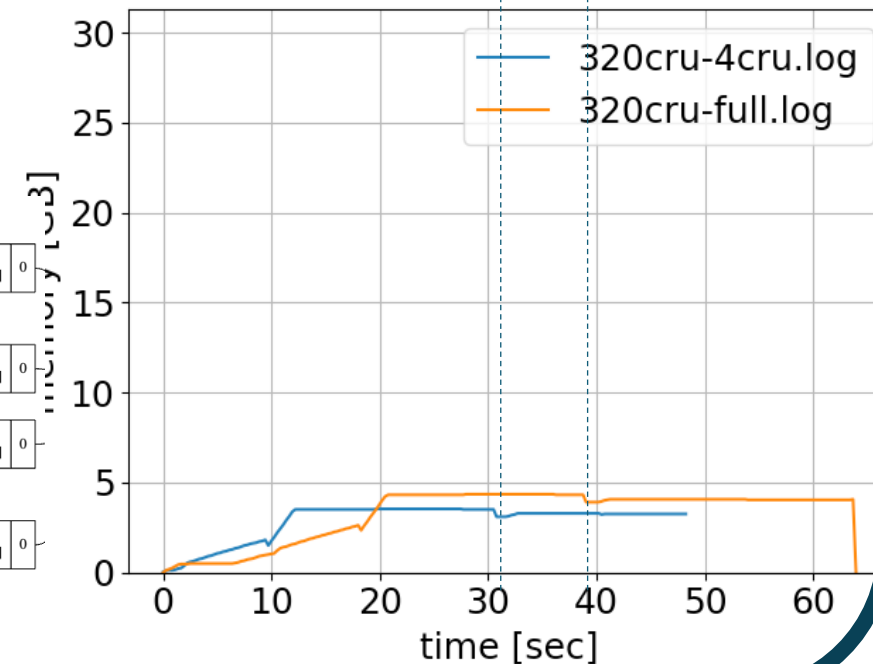
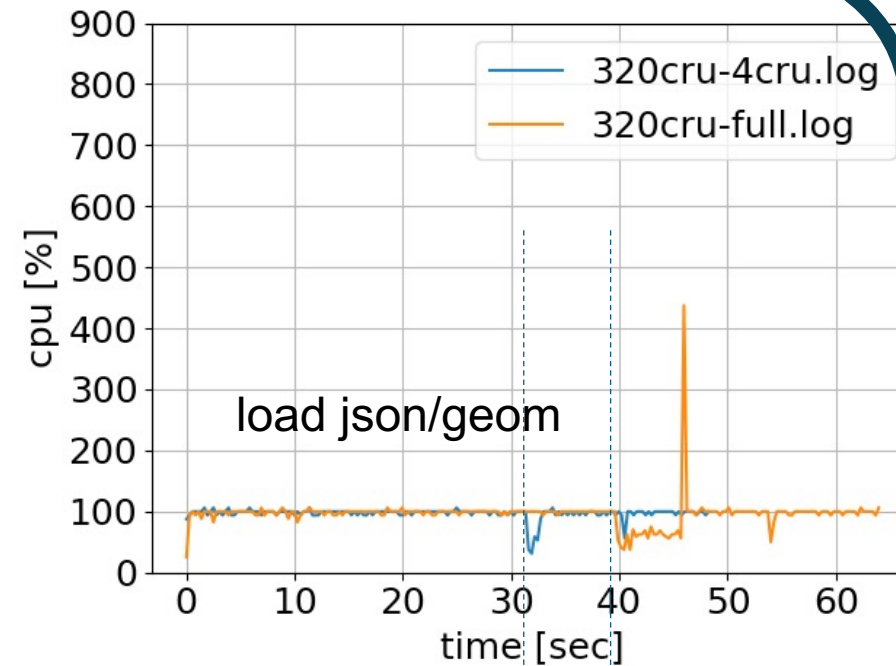
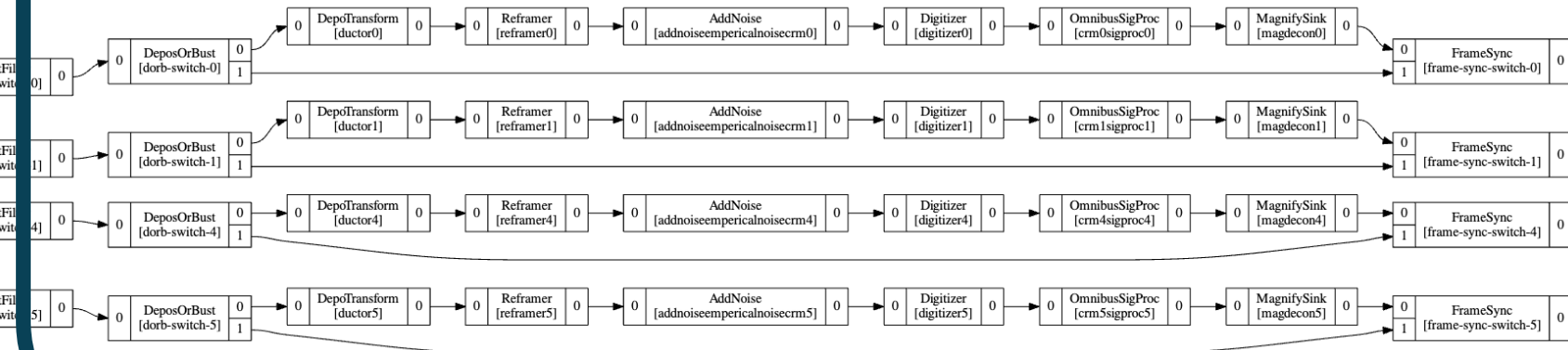
Initial tests of the Shortcuts

We were able to make async node "Hydra" working in Wire-Cell:

- <https://github.com/WireCell/wire-cell-toolkit/pull/169>
- <https://github.com/WireCell/wire-cell-toolkit/pull/271>

Initial tests for full 320CRU geom, ideal depo tracks in 0, 4

- process 4CRU: **~18sec**
- process all with shortcut: **~25sec**



New Developments between Jan. and Now

- Fixed syncing mechanism and the current 2 async. nodes (DeposOrBust, FrameSync)
 - <https://github.com/WireCell/wire-cell-toolkit/pull/271>
- Made a LArSoft integrated configuration (i.e., fcl + jsonnet) and did some tests
 - <https://github.com/HaiwangYu/hydra-skip>

3/25 results

3min to load ref jsonnet

```
4 [02:05:49.223] D [ main ] loading config file wcls-sim-drift-simchannel-nf-sp.jsonnet
5 [02:08:49.024] D [ main ] adding plugin: "WireCellPgraph"
6 [02:08:49.286] D [ sys ] loaded plugin #1 "WireCellPgraph" from library "libWireCellPgraph.so": 0x22894750
```

ref

```
30124 TrigReport ----- Event summary -----
30125 TrigReport Events total = 1 passed = 1 failed = 0
30126
30127 TrigReport ----- Modules in End-path -----
30128 TrigReport      Run      Success      Error Name
30129 TrigReport      1          1          0 out
30130
30131 TimeReport ----- Time summary [sec] -----
30132 TimeReport CPU = 4032.907430 Real = 4040.049411
30133
30134 MemReport ----- Memory summary [base-10 MB] -----
30135 MemReport VmPeak = 6211.26 VmHWM = 4751.53
30136
30137 Art has completed and will exit with status 0.
30138
30139 real    72m48.755s
30140 user    73m3.187s
30141 sys     0m6.832s
2x8x40-ref/log
```

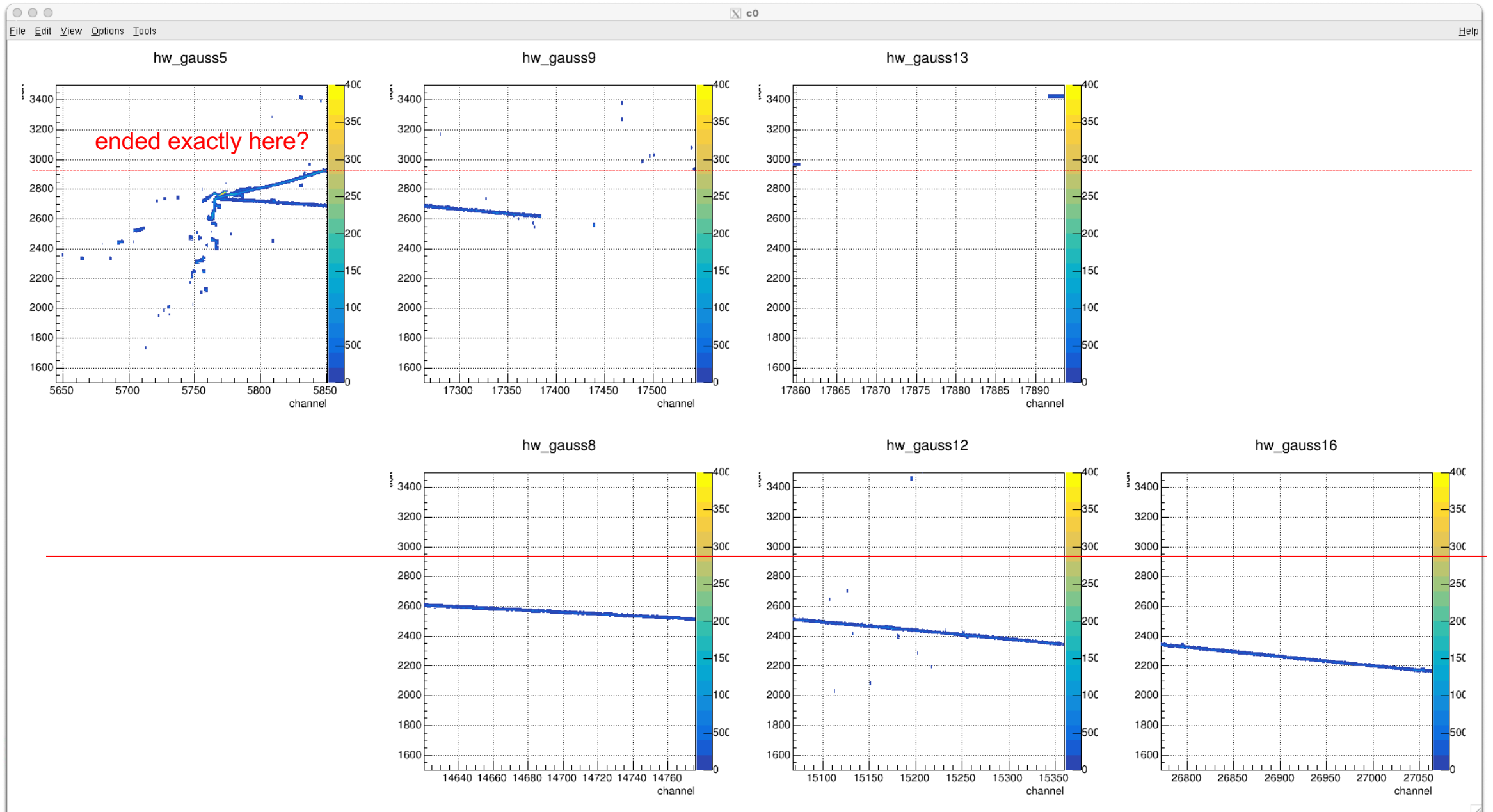
4min to load ref jsonnet

```
4 [01:20:52.045] D [ main ] loading config file wcls-sim-drift-simchannel-nf-sp.jsonnet
5 [01:24:51.235] D [ main ] adding plugin: "WireCellPgraph"
6 [01:24:51.458] D [ sys ] loaded plugin #1 "WireCellPgraph" from library "libWireCellPgraph.so": 0x49e09
```

skip

```
31564 TrigReport ----- Event summary -----
31565 TrigReport Events total = 1 passed = 1 failed = 0
31566
31567 TrigReport ----- Modules in End-path -----
31568 TrigReport      Run      Success      Error Name
31569 TrigReport      1          1          0 out
31570
31571 TimeReport ----- Time summary [sec] -----
31572 TimeReport CPU = 259.755464 Real = 261.540101
31573
31574 MemReport ----- Memory summary [base-10 MB] -----
31575 MemReport VmPeak = 6279.77 VmHWM = 4479.64
31576
31577 Art has completed and will exit with status 0.
31578
31579 real    10m49.516s
31580 user    11m34.617s
31581 sys     0m5.902s
2x8x40-skip/log
```

3/25 results, 6 CRUs



3/25 results, zoom out

Execution time (4032sec/259sec) ~ 15.5 times faster than baseline (no skip)

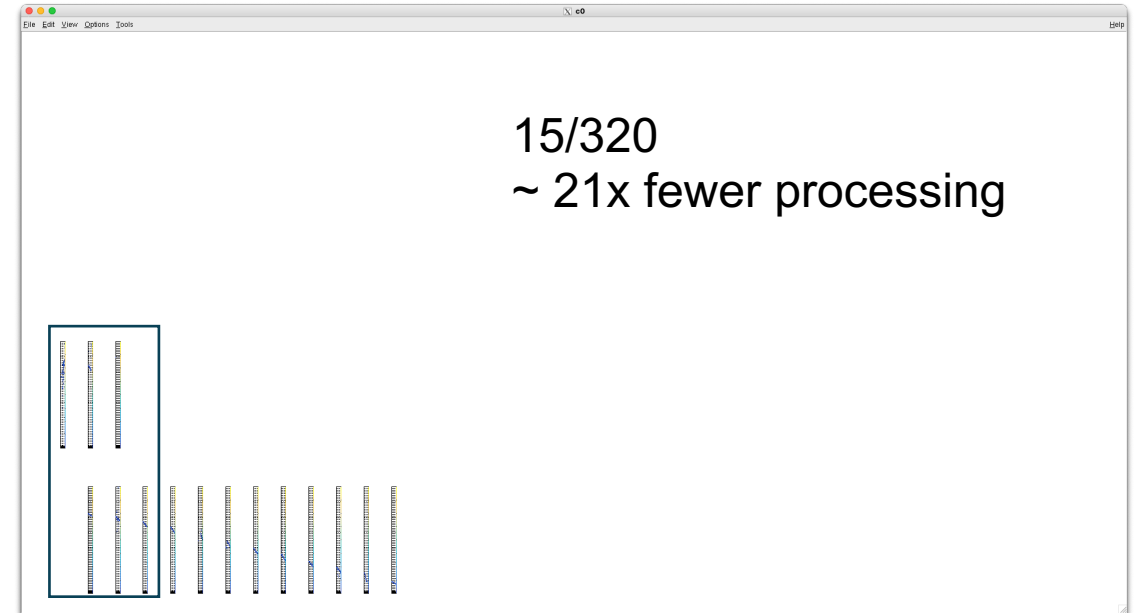
- **This ratio depends on the event activity**
- 3, 4min for compiling jsonnet, 1.5min for loading geom
 - some configure time? **still checking**

Processed CRUs (ref/skip: 320/15) ~ 21 times

- some overhead compared to 15.5, **could check this too**

A bit reduced VmHWM

Output artROOT files is 21MB



Discussions

- currently using bridged wire geom, want to continue to use no-bridged version?
 - `dunevd10kt_3view_30deg_v5_refactored_1x8x6ref.json.bz2`
- compiling jsonnet takes 3-4 min for full FD, but can be interfaced with fcl
- can we keep raw digits?
 - time: maybe OK with skip
 - memory: needs tests
 - disk: needs tests
 - root limitation: need to save to separated containers

backup

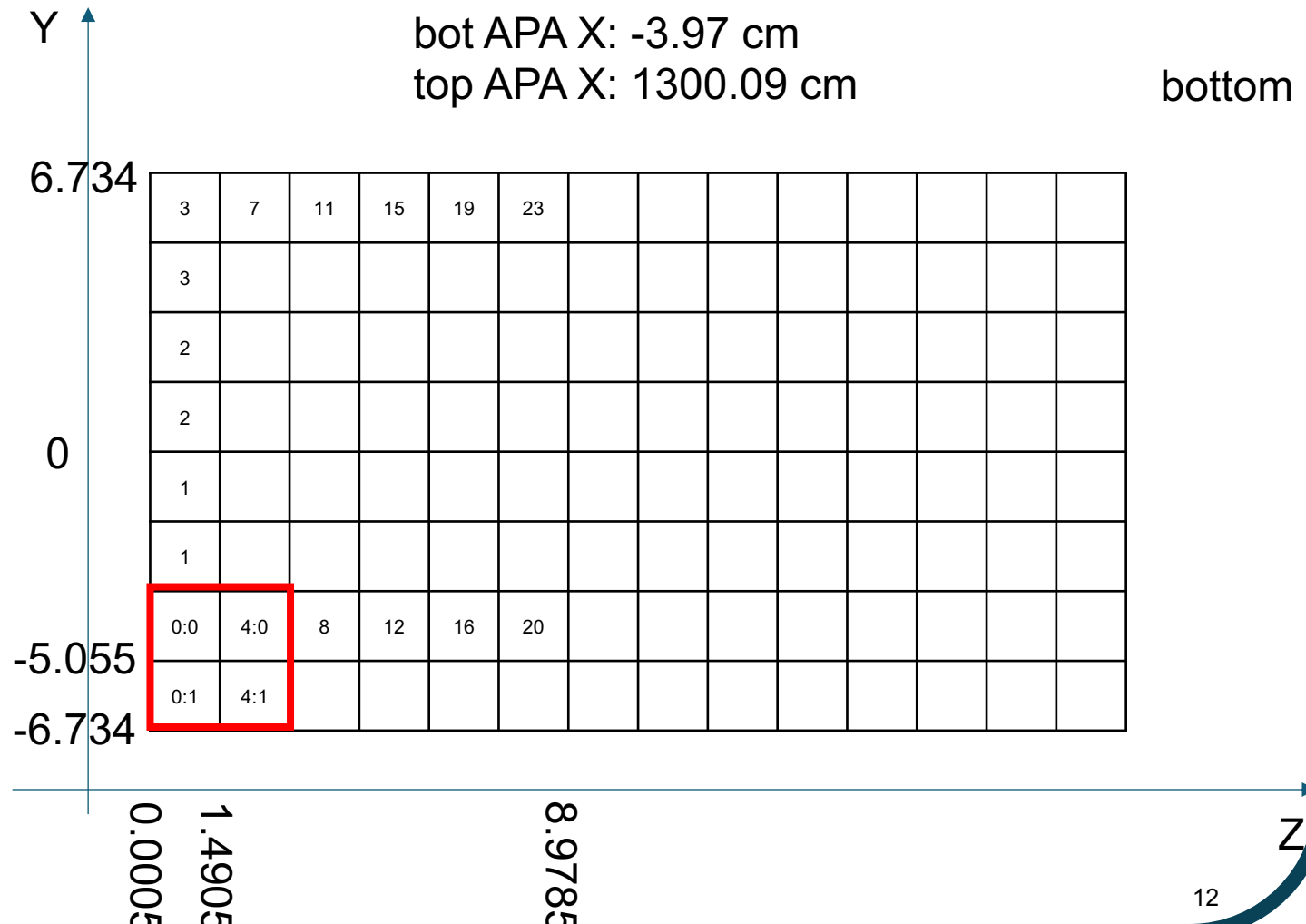
FD-VD full 10kt

- Thanks to L. Paulucci and V. Pec for the help on the wire geom. gdmls
 - Some updates needed for [DUNEGeometryHelper service](#) to use [GeoObjectSorterCRU60D](#)
- The WC wire geom file is 7MB compared to 0.55MB of 1x8x6.
- An initial WC Sim/SigProc configuration is done.

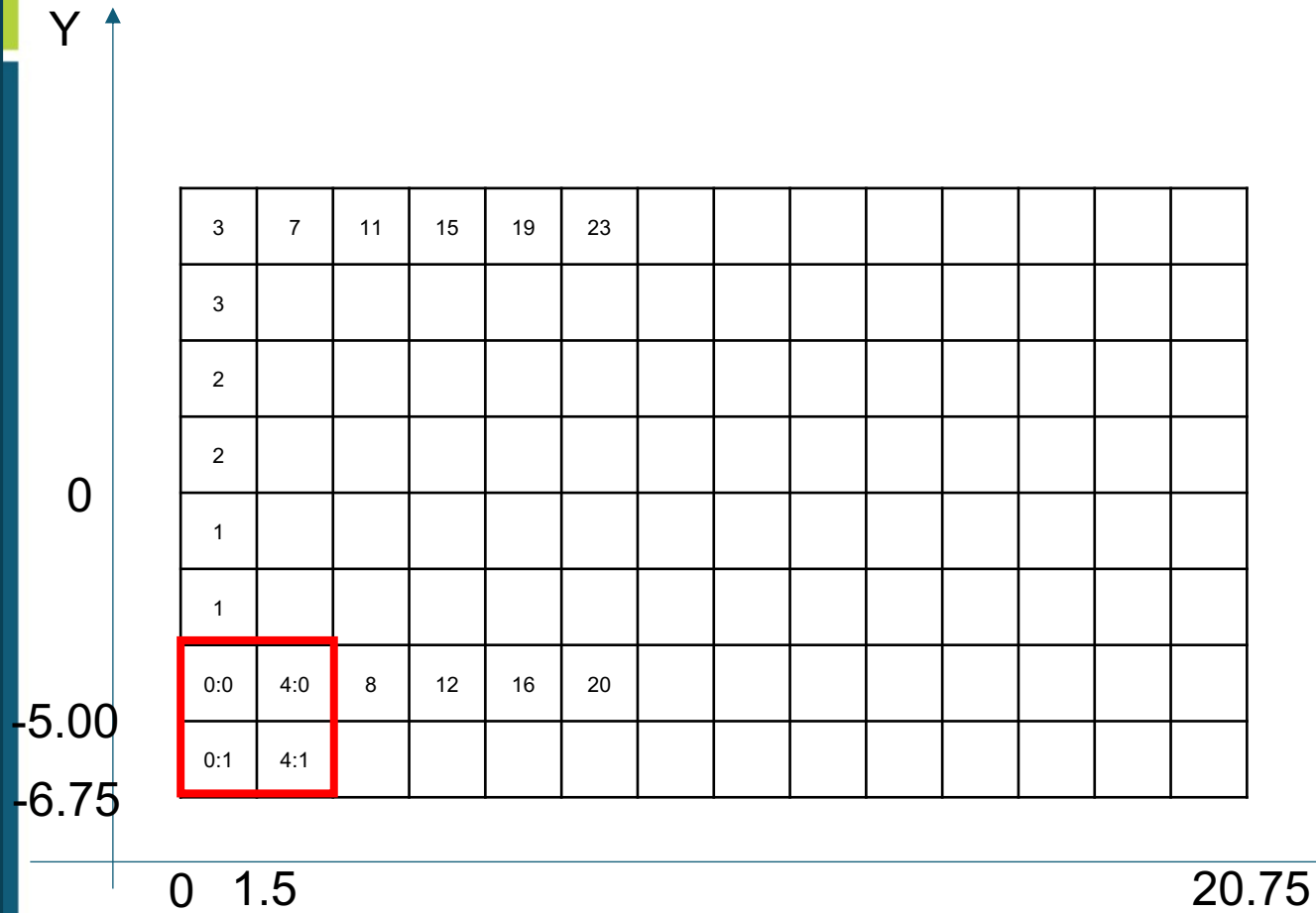
Full 10kt

$2*8(\text{CRM}, 1.5*1.5)*40$

$2*4(\text{CRU})*40 = 320 \text{ CRU}$

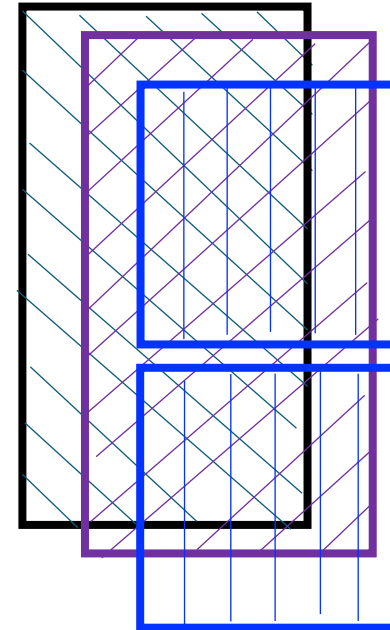


2-face numbering



CRU0

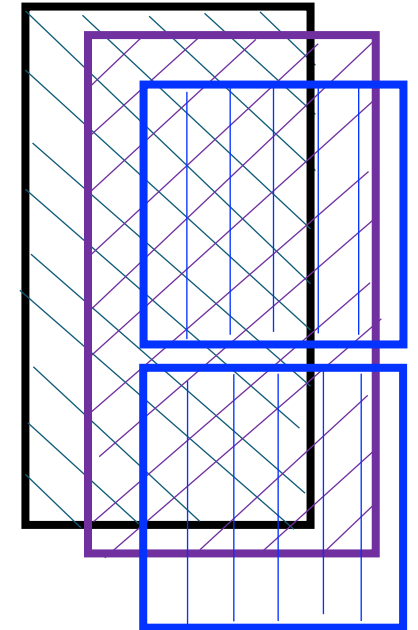
f0U: 189-475
 f0V: 1427-1142
 f0Z: 2196-2487



f1U: 0-285
 f1V: 1238-952
 f1Z: 1904-2195

CRU4

f0U: 666-951
 f0V: 1617-1903(r)
 f0Z: 2780-3071



f1U: 476-762
 f1V: 1713-1428
 f1Z: 2488-2779

3/25 results

PROCESS NAME	MODULE LABEL	PRODUCT INSTANCE NAME	DATA PRODUCT TYPE	..SIZE
GenieGen	generator		std::vector<simb::GTruth>	...1
GenieGen	TriggerResults		art::TriggerResults	...1
GenieGen	generator		std::vector<sim::BeamGateInfo>	...1
GenieGen	rns		std::vector<art::RNGsnapshot>	...0
GenieGen	generator		std::vector<simb::MCTruth>	...1
GenieGen	generator		art::Assns<simb::MCTruth, simb::MCFlux, void>	...1
GenieGen	generator		std::vector<simb::MCFlux>	...1
GenieGen	generator		art::Assns<simb::MCTruth, simb::GTruth, void>	...1
G4	IonAndScintExternal		std::vector<sim::SimEnergyDeposit>	..1844
G4	PDFastSimExternal		std::vector<sim::SimPhotonsLite>	..168
G4	elecDrift		std::vector<sim::SimChannel>	..6493
G4	PDFastSimExternal		std::vector<sim::OpDetBacktrackerRecord>	..156
G4	PDFastSimAr		std::vector<sim::SimPhotonsLite>	..168
G4	rns		std::vector<art::RNGsnapshot>	...10
G4	IonAndScint		std::vector<sim::SimEnergyDeposit>	..61993
G4	TriggerResults		art::TriggerResults	...1
G4	PDFastSimAr		std::vector<sim::OpDetBacktrackerRecord>	..168
G4	largeant	LArG4DetectorServicevolExternalActive	std::vector<sim::SimEnergyDeposit>	..1844
G4	largeant		std::vector<simb::MCParticle>	..280
G4	largeant	LArG4DetectorServicevolTPCActive	std::vector<sim::SimEnergyDeposit>	..61993
G4	PDFastSimXe		std::vector<sim::OpDetBacktrackerRecord>	..168
G4	largeant		std::map<int, std::set<int> >	...86
G4	PDFastSimXe		std::vector<sim::SimPhotonsLite>	..168
G4	largeant		art::Assns<simb::MCTruth, simb::MCParticle, sim::GeneratedParticleInfo>	...280
wclssim	tpcrawdecoder	simpleSC	std::vector<sim::SimChannel>	491520
wclssim	TriggerResults		art::TriggerResults	...1
wclssim	tpcrawdecoder	gauss	std::vector<recob::Wire>	491520
wclssim	tpcrawdecoder	dnnsp	std::vector<recob::Wire>	491520
wclssim	tpcrawdecoder	wiener	std::vector<recob::Wire>	491520
wclssim	plopper	bogus	std::vector<sim::SimEnergyDeposit>	..713

documents

dunegpvm:

```
/Users/yuhw/wire-cell/dune-fd/hydra-skip  
/exp/dune/app/users/yuhw/dunefd/depo-gen/g4.root
```

email:

[WireCell/wire-cell-toolkit] Works for one cfg (PR #271)

Async. in WCT

FanoutCat

function
1->1

split_node
1->n

sequencer_node

sequencer_node

...

FaninCat

join_node
n->1

function
1->1

sequencer_node

HydraCat

indexer
n->1

function
1->1

multifunction
1->n

sequencer_node

sequencer_node

...

```

{
    auto* dxn = new indexer_node<Nin>(graph);
    nodes.push_back(dxn);
    receivers = indexer_ports(*dxn, std::make_index_sequence<Nin>{});
    auto* fn = new tbb::flow::function_node<indexer_msg_t<Nin>, tagged_msg_t>(graph, 1, HydraInputBody<Nin>);
    nodes.push_back(fn);
    make_edge(*dxn, *fn);
}

{
    auto mfn = new mfunc_node_type<Nout>(graph, 1, HydraOutputBody<Nout>(wcnode, info));
    nodes.push_back(mfn);
    auto spv = outdexter_ports(*mfn, std::make_index_sequence<Nout>{});
    for (size_t ind=0; ind<Nout; ++ind) {
        auto qn = new seq_node(graph, [] (const msg_t& m) {return m.first;});
        nodes.push_back(qn);
        tbb::flow::make_edge(*spv[ind], *qn);
        senders.push_back(dynamic_cast<sender_type*>(qn));
    }
}

```

indexer
n->1

function
1->1

multifunction
1->n

sequencer_node

sequencer_node

...