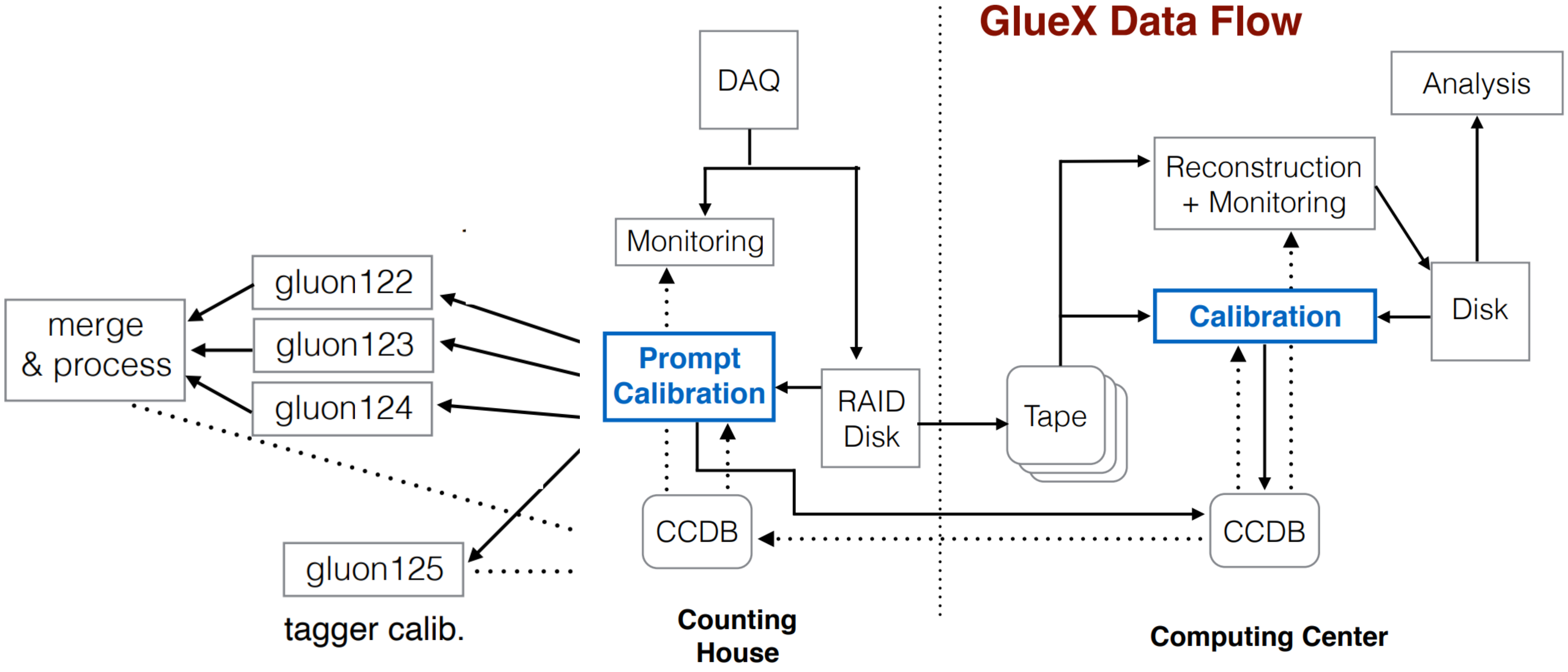
Jon Zarling

Calibration Workflows with Cylc

10/2/2024

GlueX Calibrations

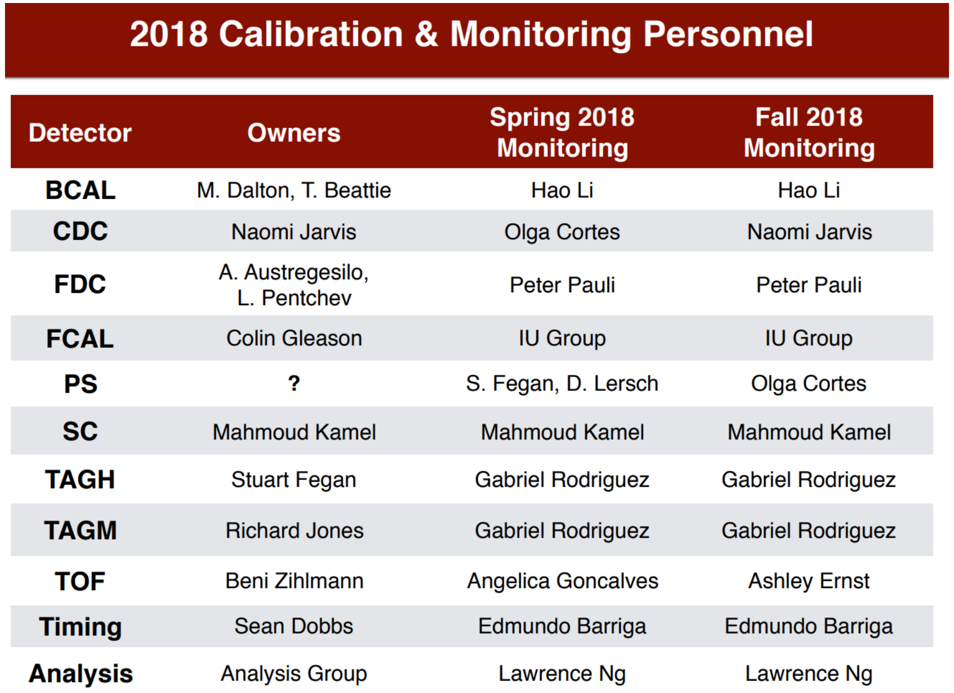


https://halldweb.jlab.org/doc-private/DocDB/ShowDocument?docid=4209

Calibrations are a complicated, multi-step challenge

Calibration Workflow

Detector	Procedures
BCAL	None
CDC	Gains, could add ToTD
FDC	Base, wire timing
FCAL	Base time, could add others
PS	Base, channel timing, timewalks
SC	Base, channel timing
TAGH	Base, channel timing, timewalks
TAGM	Base, channel timing, timewalks
TOF	Base timing, rest done via skims



2018 Calibration & Monitoring Personnel

Detector	Owners	Spring 2018 Monitoring	Fall 2018 Monitoring
BCAL	M. Dalton, T. Beattie	Hao Li	Hao Li
CDC	Naomi Jarvis	Olga Cortes	Naomi Jarvis
FDC	A. Austregesilo, L. Pentchev	Peter Pauli	Peter Pauli
FCAL	Colin Gleason	IU Group	IU Group
PS	?	S. Fegan, D. Lersch	Olga Cortes
SC	Mahmoud Kamel	Mahmoud Kamel	Mahmoud Kamel
TAGH	Stuart Fegan	Gabriel Rodriguez	Gabriel Rodriguez
TAGM	Richard Jones	Gabriel Rodriguez	Gabriel Rodriguez
TOF	Beni Zihlmann	Angelica Goncalves	Ashley Ernst
Timing	Sean Dobbs	Edmundo Barriga	Edmundo Barriga
Analysis	Analysis Group	Lawrence Ng	Lawrence Ng

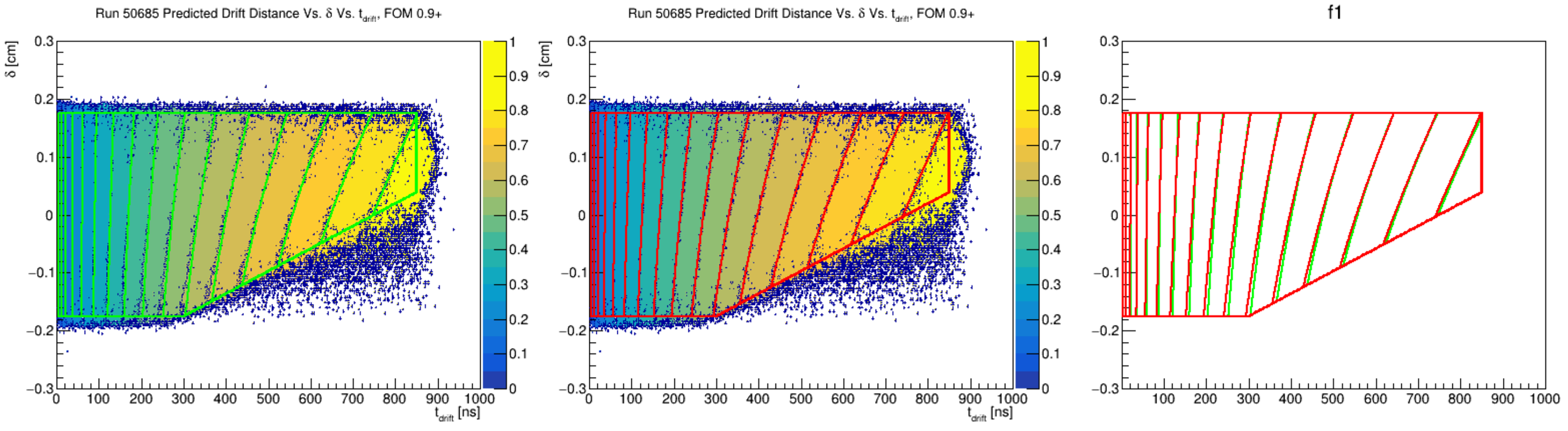
https://halldweb.jlab.org/doc-private/DocDB/ShowDocument?docid=4209

- Calibrations are complicated, multi-step challenge
- Subsystems usually spread out across many people
  ↳ sometimes people can work in parallel, not always
- ***Wouldn't it be nice to bundle this up more?***

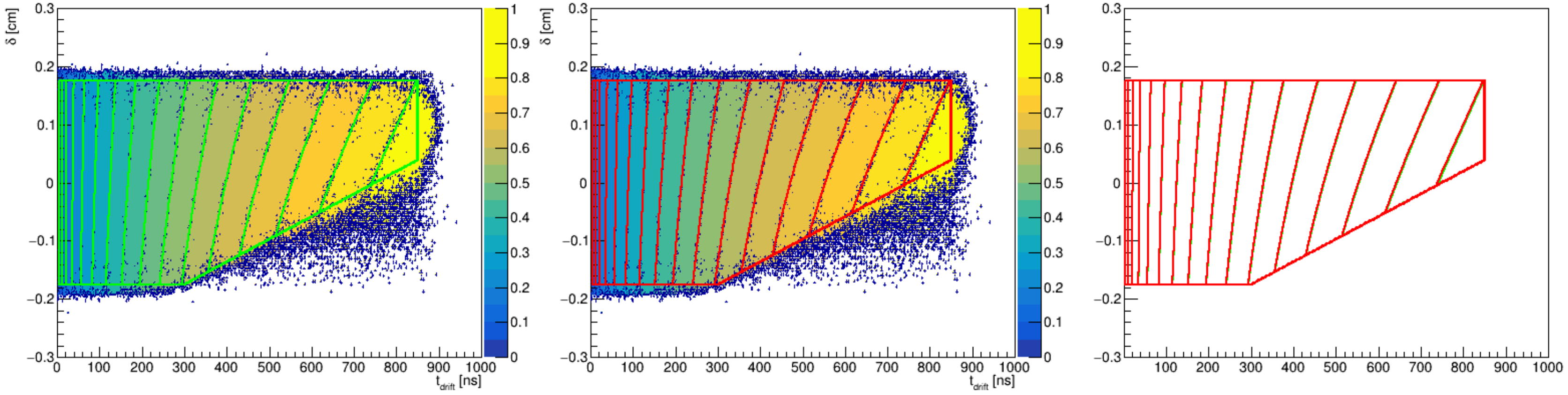
Case Study: CDC Time-to-Distance (TTOD)

Sometimes procedure need to iterate/converge, too

Here: converged when red totally lies on top of green



Red: 0th iteration (pre-calib)
Green: 1st iteration

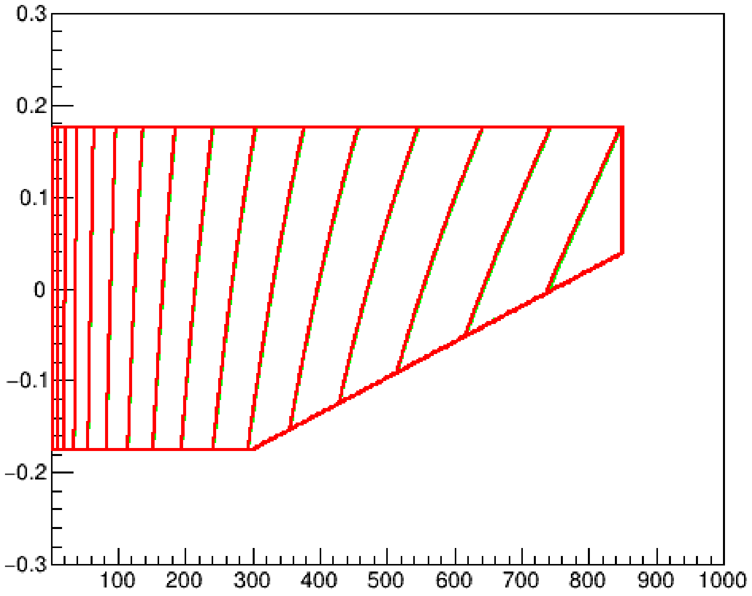


Red: 1st iteration
Green: 2nd iteration

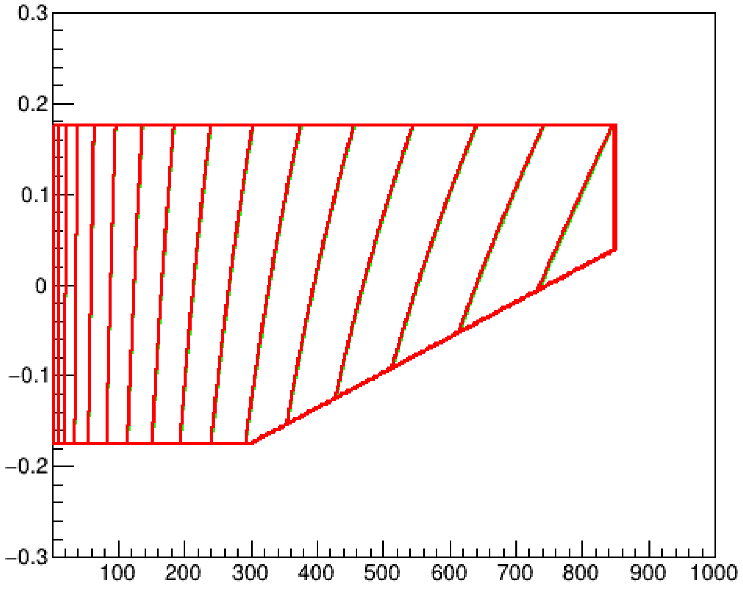
Case Study: CDC Time-to-Distance (TTOD)

Here: converged when red totally lies on top of green

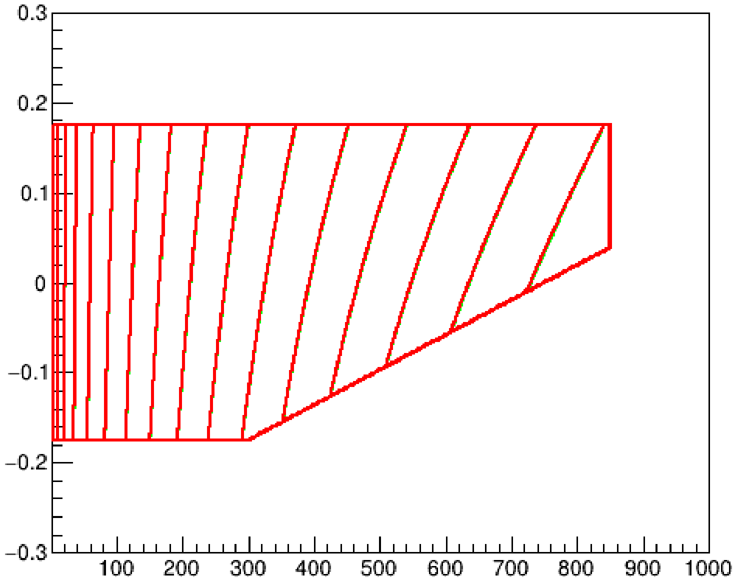
Red: 1st iteration
Green: 2nd iteration



Red: 2nd iteration
Green: 3rd iteration



Red: 7th iteration
Green: 8th iteration



Red/green comparison can be quantified in terms of CDC resolution (for later)

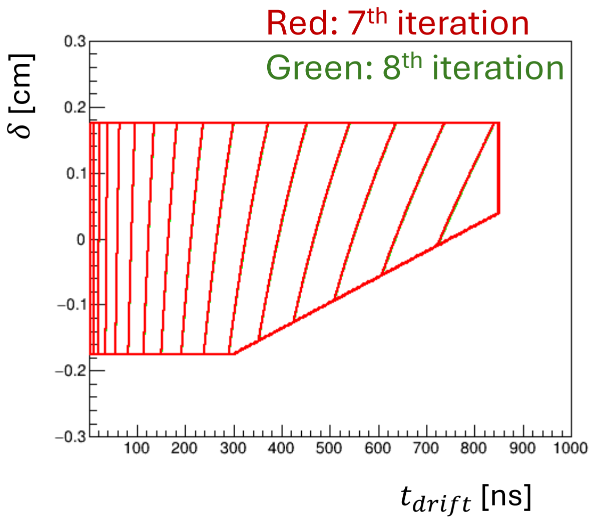
Calibration Workflows with cylc

(pronounced “silk”)

Punchline: I did all this from a single terminal command!

```
> cylc vip gx_ttod
```

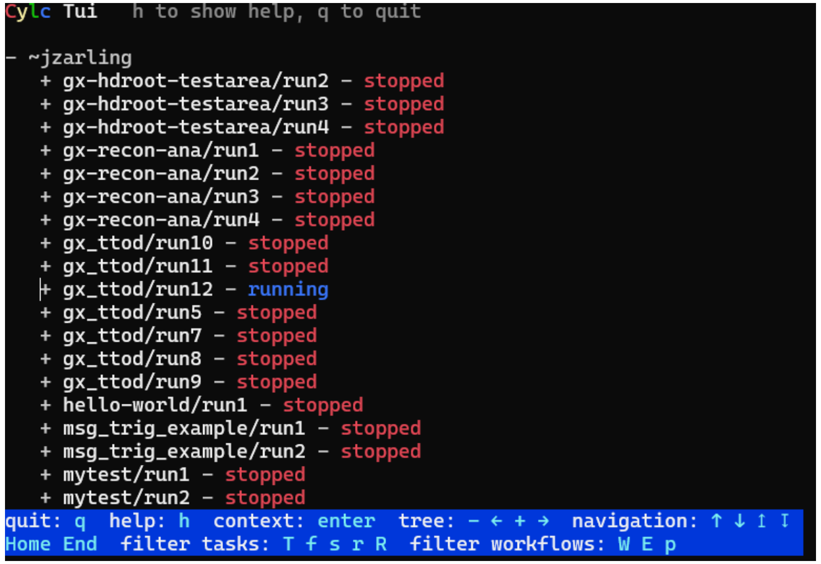
- Runs 20 jobs per iteration
- Repeats ×8 iterations
- Read/write ccdb (local copy)



Calibration Workflows with cylc

Terminal interface (interactive!):

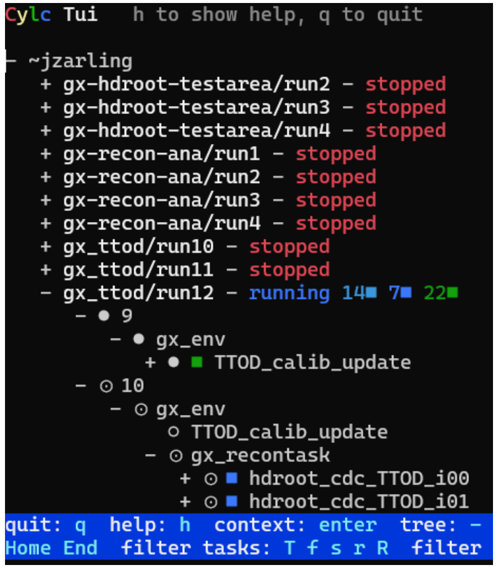
```
> cylc tui
```



```
Cylc Tui   h to show help, q to quit

- ~jzarling
   + gx-hdroot-testarea/run2 - stopped
   + gx-hdroot-testarea/run3 - stopped
   + gx-hdroot-testarea/run4 - stopped
   + gx-recon-ana/run1 - stopped
   + gx-recon-ana/run2 - stopped
   + gx-recon-ana/run3 - stopped
   + gx-recon-ana/run4 - stopped
   + gx_ttod/run10 - stopped
   + gx_ttod/run11 - stopped
   + gx_ttod/run12 - running
   + gx_ttod/run5 - stopped
   + gx_ttod/run7 - stopped
   + gx_ttod/run8 - stopped
   + gx_ttod/run9 - stopped
   + hello-world/run1 - stopped
   + msg_trig_example/run1 - stopped
   + msg_trig_example/run2 - stopped
   + mytest/run1 - stopped
   + mytest/run2 - stopped
quit: q  help: h  context: enter  tree: - ← + →  navigation: ↑ ↓ ↥ ↧
Home End  filter tasks: T f s r R  filter workflows: W E p
```

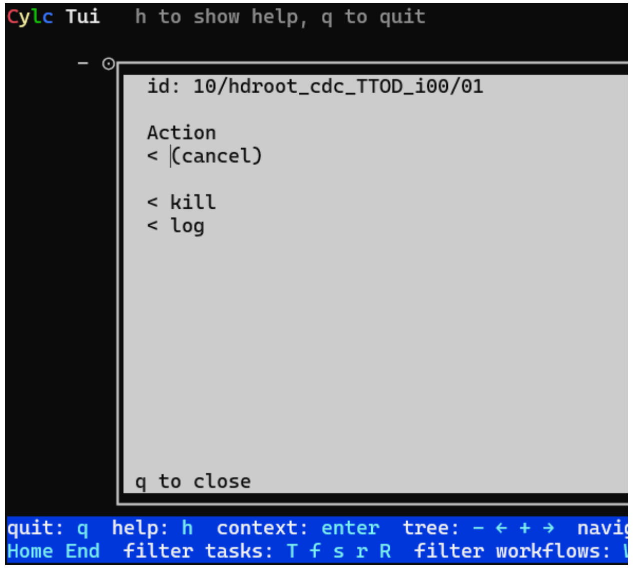
Mouse click in terminal to expand



```
Cylc Tui   h to show help, q to quit

- ~jzarling
   + gx-hdroot-testarea/run2 - stopped
   + gx-hdroot-testarea/run3 - stopped
   + gx-hdroot-testarea/run4 - stopped
   + gx-recon-ana/run1 - stopped
   + gx-recon-ana/run2 - stopped
   + gx-recon-ana/run3 - stopped
   + gx-recon-ana/run4 - stopped
   + gx_ttod/run10 - stopped
   + gx_ttod/run11 - stopped
   - gx_ttod/run12 - running 14■ 7■ 22■
      - ● 9
         - ● gx_env
            + ● ■ TTOD_calib_update
      - ⊙ 10
         - ⊙ gx_env
              ○ TTOD_calib_update
            - ⊙ gx_recontask
               + ⊙ ■ hdroot_cdc_TTOD_i00
               + ⊙ ■ hdroot_cdc_TTOD_i01
quit: q  help: h  context: enter  tree: -
Home End  filter tasks: T f s r R  filter
```

Individual or group job interventions



```
Cylc Tui   h to show help, q to quit

      - ⊙
           id: 10/hdroot_cdc_TTOD_i00/01

           Action
           < (cancel)

           < kill
           < log









          q to close

quit: q  help: h  context: enter  tree: - ← + →  navi
Home End  filter tasks: T f s r R  filter workflows: W
```

(also has cool GUI version)

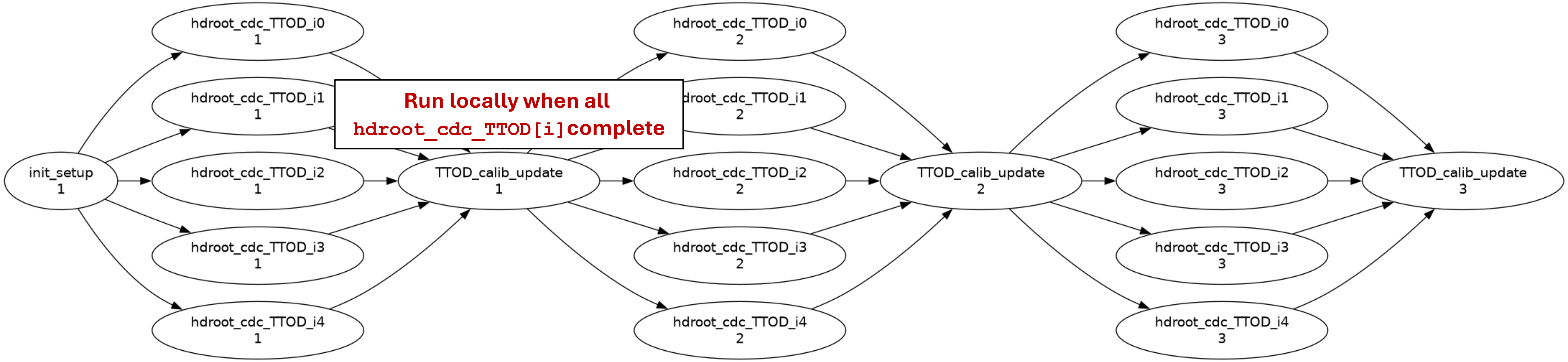
What is Cylc?

Cylc is a decentralised, distributed, DAG/DCG scheduler.

Workflows can be written as graphs

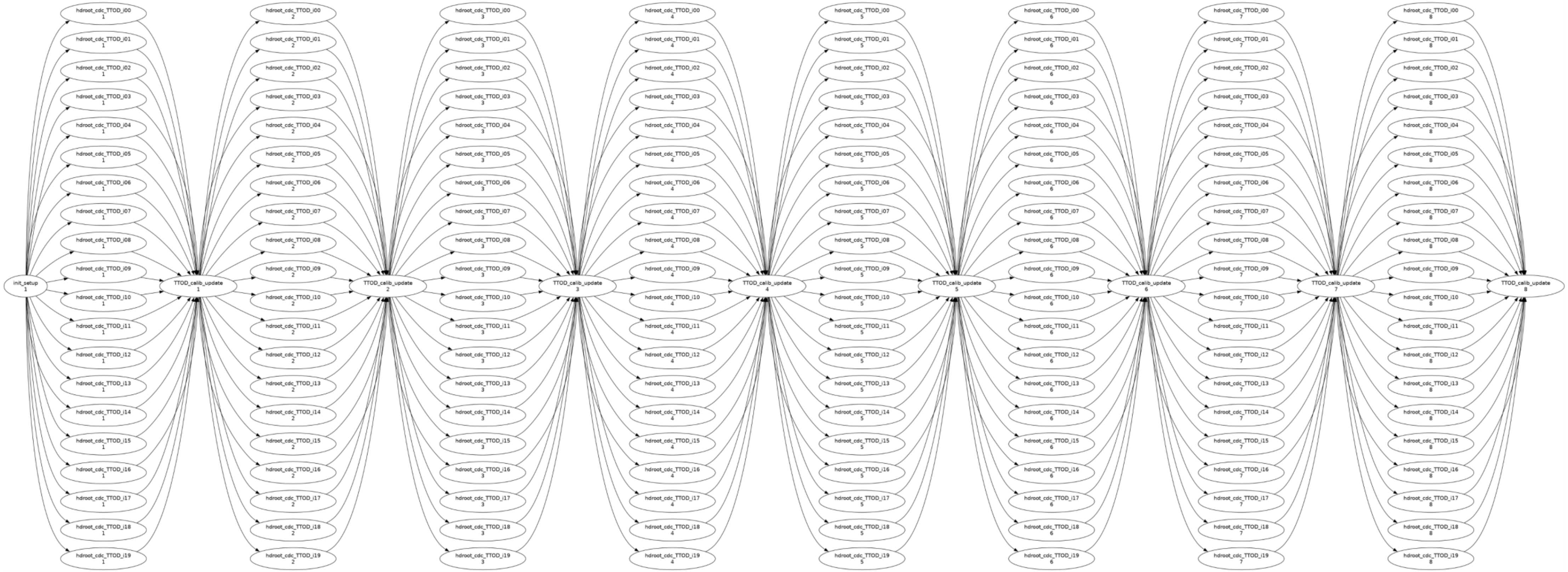
- Example: time-to-distance calibrations (shortened to 5 files, 3 iterations)
- Each bubble is a separate task to run
- Task = slurm farm job or local script

Submit jobs when `init_setup[1]` completes



Run locally when all `hdroot_cdc_TTOD[i]` complete

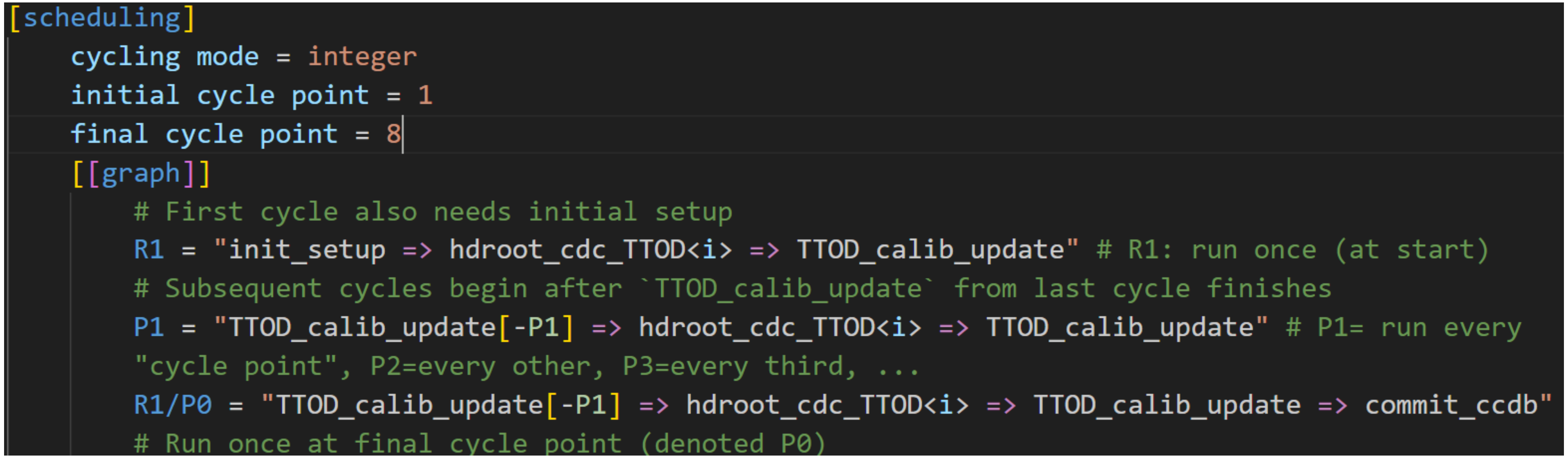
Full Graph, CDC Time-to-Distance Calib.



Defining Workflows

Workflows described in file called flow.cylc

- Gives graph of tasks to run

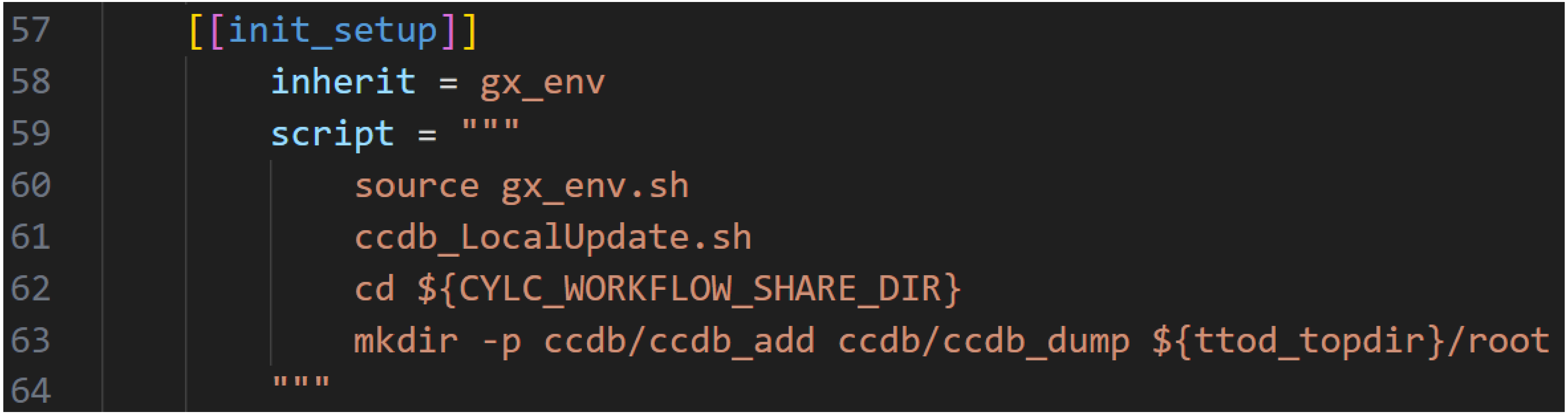


```
[scheduling]
    cycling mode = integer
    initial cycle point = 1
    final cycle point = 8
    [[graph]]
        # First cycle also needs initial setup
        R1 = "init_setup => hdroot_cdc_TTOD<i> => TTOD_calib_update" # R1: run once (at start)
        # Subsequent cycles begin after `TTOD_calib_update` from last cycle finishes
        P1 = "TTOD_calib_update[-P1] => hdroot_cdc_TTOD<i> => TTOD_calib_update" # P1= run every
        "cycle point", P2=every other, P3=every third, ...
        R1/P0 = "TTOD_calib_update[-P1] => hdroot_cdc_TTOD<i> => TTOD_calib_update => commit_ccdb"
        # Run once at final cycle point (denoted P0)
```

Defining Workflows, cont.

Workflows described in file called flow.cylc

A single “task”

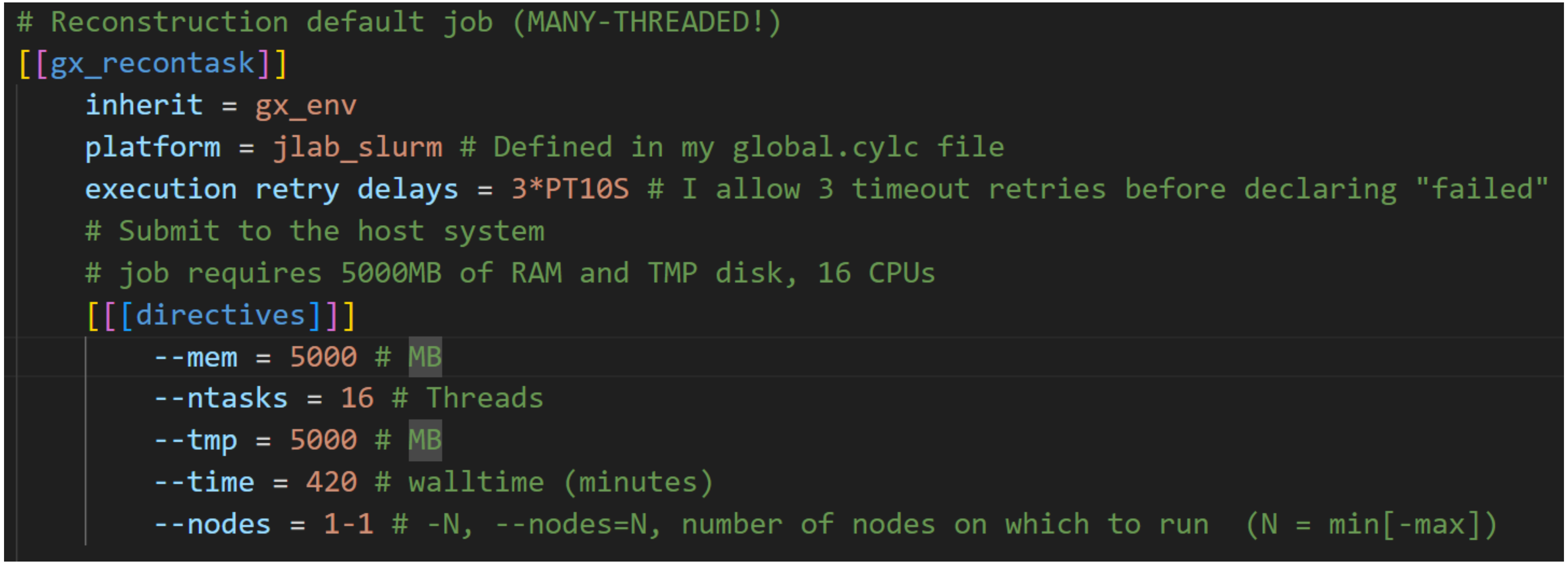


```
    [[init_setup]]
        inherit = gx_env
        script = """
            source gx_env.sh
            ccdb_LocalUpdate.sh
            cd ${CYLC_WORKFLOW_SHARE_DIR}
            mkdir -p ccdb/ccdb_add ccdb/ccdb_dump ${ttod_topdir}/root
        """
```

Defining Workflows, cont.

Workflows described in file called flow.cylc

Defining how to run batch jobs

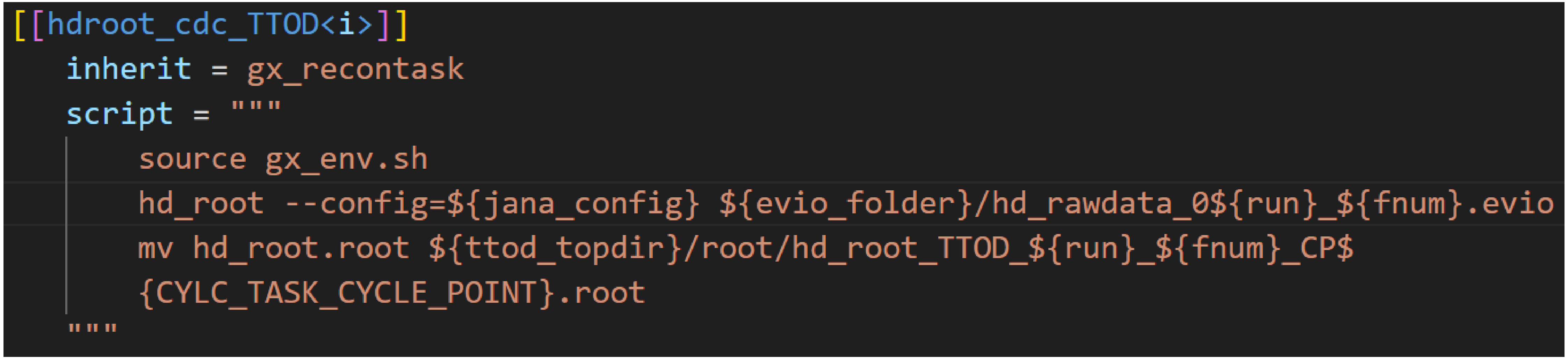


```
# Reconstruction default job (MANY-THREADED!)
[[gx_recontask]]
    inherit = gx_env
    platform = jlab_slurm # Defined in my global.cylc file
    execution retry delays = 3*PT10S # I allow 3 timeout retries before declaring "failed"
    # Submit to the host system
    # job requires 5000MB of RAM and TMP disk, 16 CPUs
    [[[directives]]]
        --mem = 5000 # MB
        --ntasks = 16 # Threads
        --tmp = 5000 # MB
        --time = 420 # walltime (minutes)
        --nodes = 1-1 # -N, --nodes=N, number of nodes on which to run  (N = min[-max])
```

Defining Workflows, cont.

Workflows described in file called flow.cylc

Add `inherit = gx_recontask` to run as batch job
(otherwise runs locally)



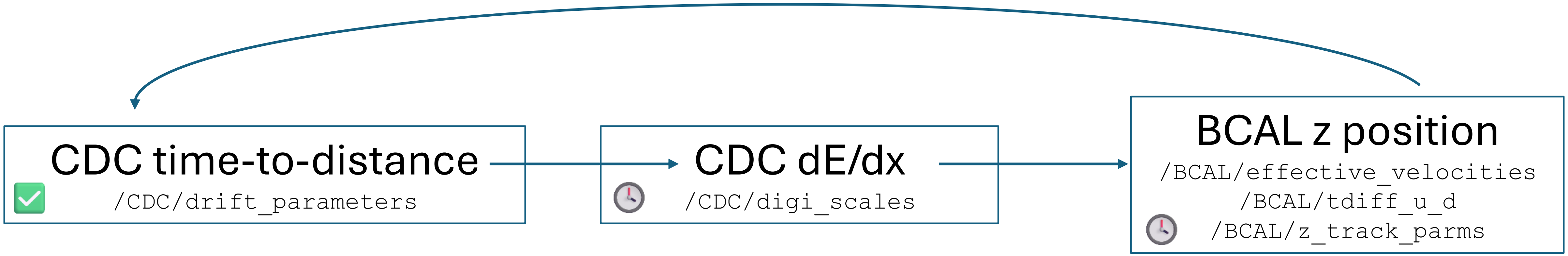
```
[[hdroot_cdc_TTOD<i>]]
   inherit = gx_recontask
   script = """
       source gx_env.sh
       hd_root --config=${jana_config} ${evio_folder}/hd_rawdata_0${run}_${fnum}.evio
       mv hd_root.root ${ttod_topdir}/root/hd_root_TTOD_${run}_${fnum}_CP$
       {CYLC_TASK_CYCLE_POINT}.root
   """
```

Calibration Step
[CCDB/table_modified]

Looking Ahead

Will iterating over multiple subsystems improve calibrations?
(for some figure-of-merit we define)

GLOBAL CALIBRATION LOOP



CDC time-to-distance
✅ /CDC/drift_parameters

CDC dE/dx
🕓 /CDC/digi_scales

BCAL z position
/BCAL/effective_velocities
/BCAL/tdiff_u_d
🕓 /BCAL/z_track_parms

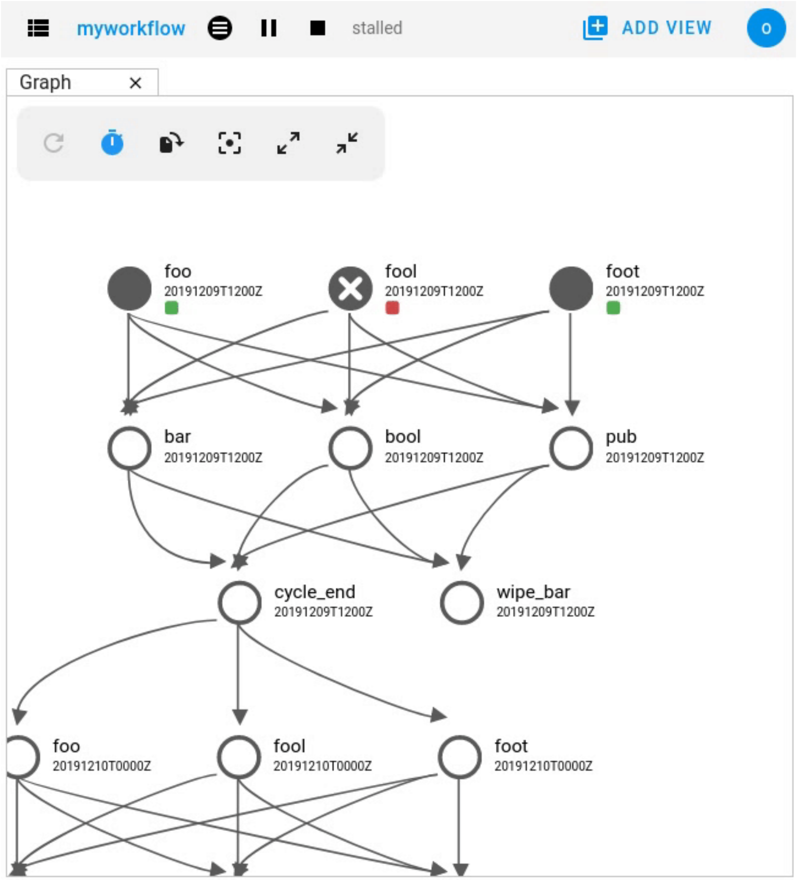
FOM: CDC track resolution / BCAL z-vertex residuals

+ FDC?
+ Start Counter?

Looking Farther Ahead?

Web GUI can be deployed via Jupyter Hub setup:

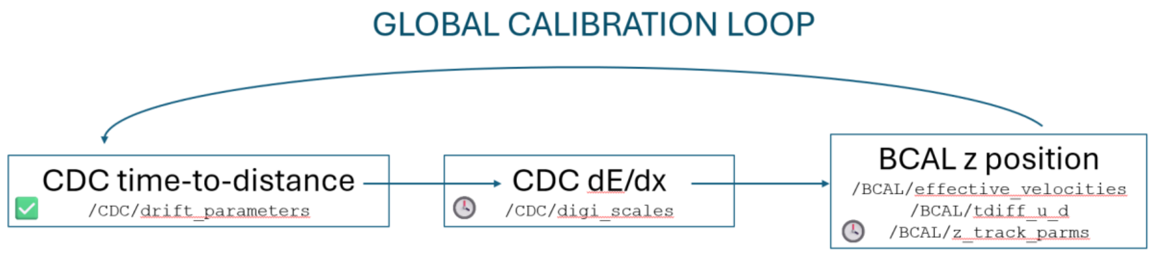
- Multiple users can access, monitor, modify, etc.
- Different users can have different privilege levels
  - Person A: global control
  - Person B: can start/stop/modify jobs related to their subdetector
  - Person C: can look at monitoring plots



Summary

- Cylc looks like a great tool for handling calibration workflows
- Simple demonstration with CDC today

- Discussion point: other loops over subsystems to try?



GLOBAL CALIBRATION LOOP

CDC time-to-distance
✓ /CDC/drift_parameters

CDC dE/dx
/CDC/digi_scales

BCAL z position
/BCAL/effective_velocities
/BCAL/tdiff_u_d
/BCAL/z_track_parms