Software Working Group Session

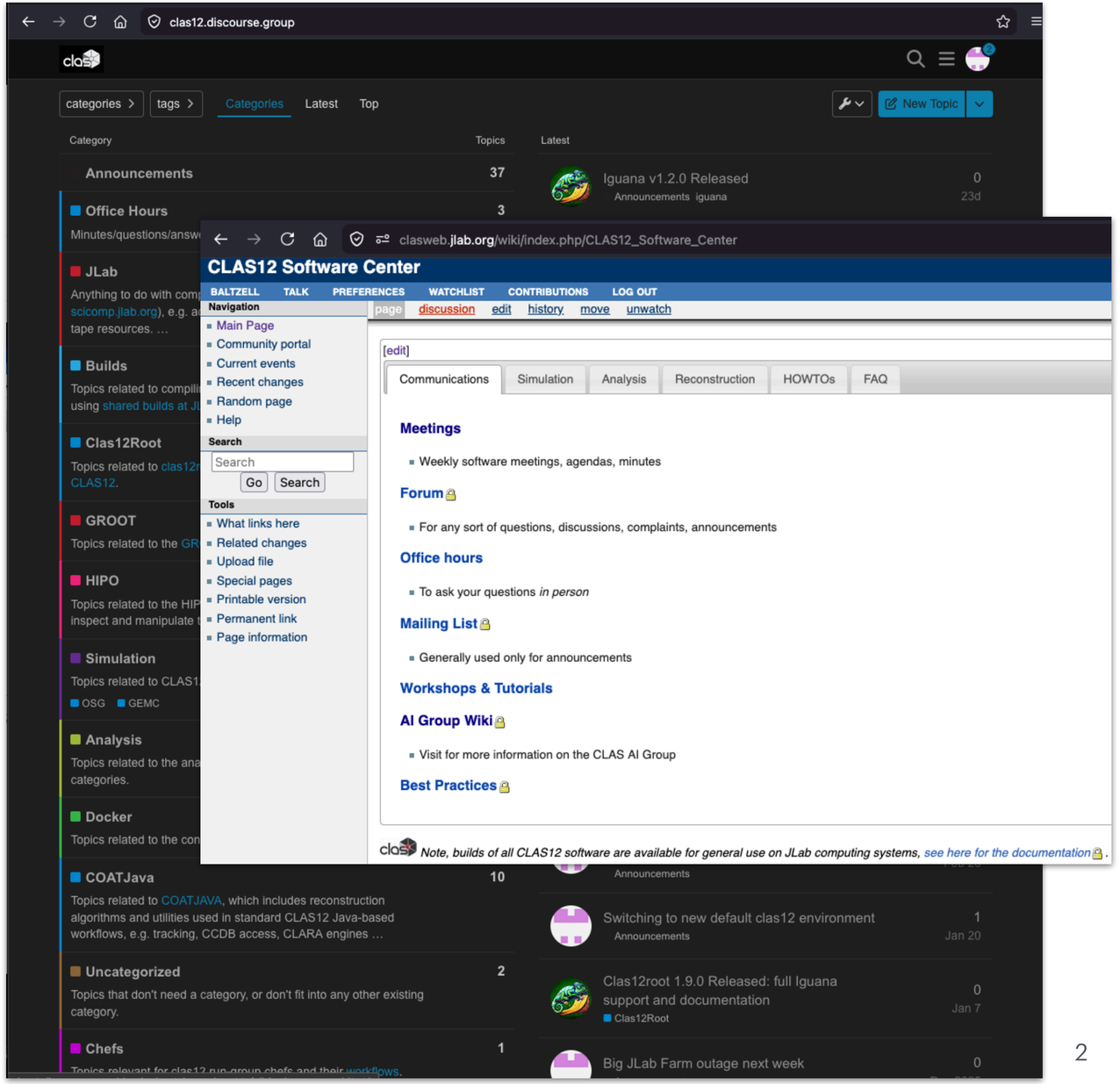
CLAS Collaboration Meeting

Nathan Baltzell - June 29, 2026

Introduction

Software Working Group

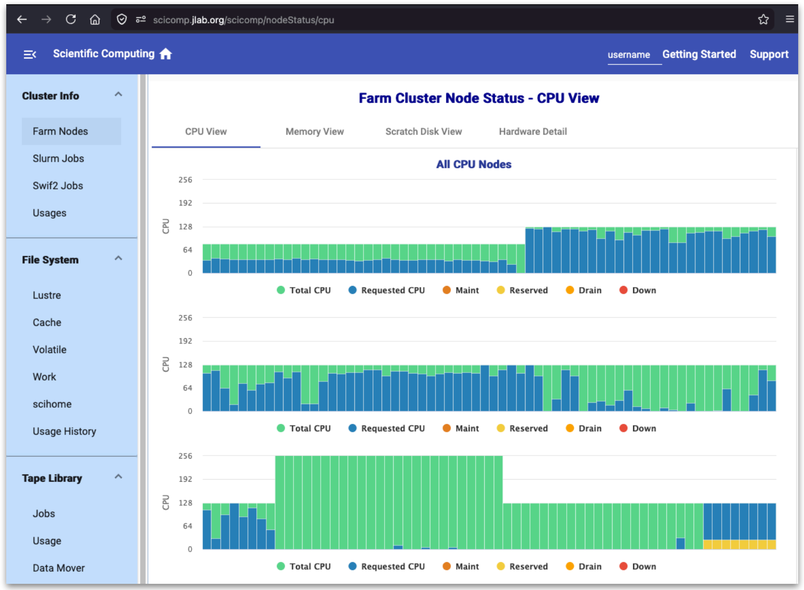
- Main page, documentation central: https://clasweb.jlab.org/wiki/index.php/CLAS12_Software_Center
- Discourse forum: https://clas12.discourse.group/
- Weekly meetings Thursdays @ 11:00, see wiki for details
    - Once per month is AI-focused, with Richard Tyson (UGlasgow)
- Office Hours: Tuesday @ 9:00, with Raffaella De Vita
- '26 Service Work Committee, SWWG: David Heddle (CNU) and Matt Nicol (USC)
- '26 Speaker's Committee, SWWG: Pierre Chatagnon (IPNO) and Richard Tyson (UGlasgow)

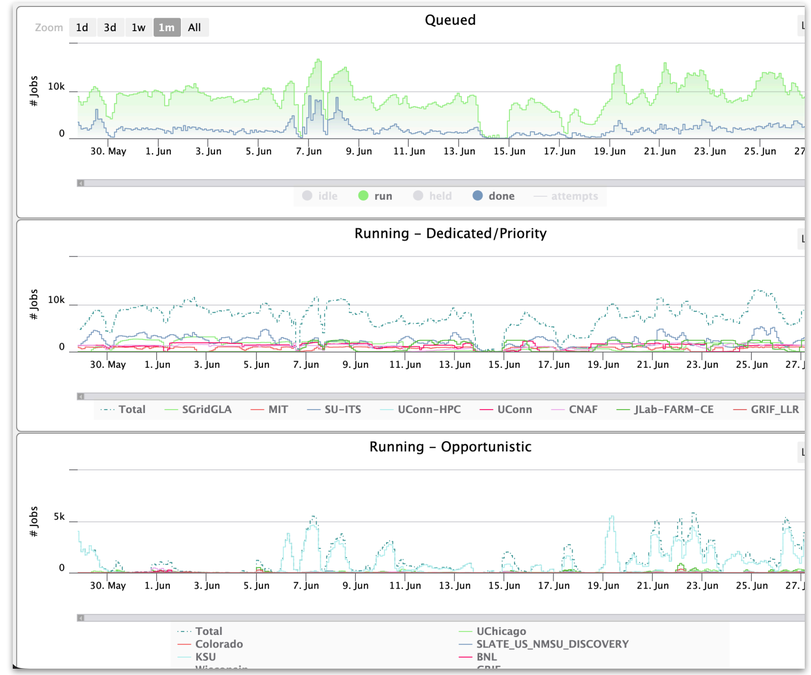


Computing Updates

- RG-E/K processing went smoothly in Spring '26 at JLab, 80-billion events, 1.5 PB
- clas12 simulations also still running smoothly on OSG.
- JLab farm changes since the last collaboration meeting
  - farm25 nodes brought online
  - Equal CPU/thread fairshare billing restored
  - A:B:C:D:E fairshare changed, due to usage patterns, 26:26:8:26:14 → 25:30:8:30:7

https://scicomp.jlab.org





Software Updates

- Gemc and Coatjava now capable of Real-Run-Number Simulations, RG-L/ALERT/AHDC/ATOF upgrades, RG-H recoil detector and mRWell URWT merged, mCLAS12 detector available
- Coatjava speedups, scaling, profiling improvements; Decoding now CLARA-fied and multi-threaded, in use by RG-L
- HIPO/C++ library multi-threading upgraded, Iguana PID and correction algorithm additions and thread safety, mcdj for user-friendly running of the simulation pipeline
- Various documentation upgrades, software wiki next …

2026.5

- *Move to CCDB v2 schema and software*
- *Leverage real-run-number simulations, OSG submit portal updates; Leverage workflow improvements offsite*
- *RICH integration in EB/DSTs, Fast MC, kinematic fitting, …*

	+/- Track Inefficiency (%) per nA	
	Pass 2	Now
RG-A	0.143 / 0.140	0.077 / 0.074
RG-B	0.225 / 0.199	0.106 / 0.100
RG-D	0.167 / 0.159	0.079 / 0.087

Forward tracking

- µRwell+DC framework is final and comprehensively upgraded
- DC-only inefficiency <0.1%/nA, ~0.1%/nA with FT, resolution and speed improvements
- *Some digitization, resolution, covariance tuning work remains …*

Real Run Number Simulations

What and Why

- Stop simulating with run=11 and CCDB variations for geometry …
  - CLARA provides only two user hooks, the once-called `init(yaml)`, and the multi-threaded `process(event)`, and we've historically loaded geometry in the former …
- Instead, load CCDB geometry from the default variation with the same, real, run numbers as real data!
  - This requires waiting until the run number is known, in `process(event)`, so COATJAVA engines now have a new `detectorChanged(run)` hook.
- Using CCDB variations created a lot of overhead, maintenance, confusion, proliferation of config files, bookkeeping needs!
  - For experts and novices alike, both Coatjava and Gemc, for any simulation jobs !
- Using CCDB run=11 promoted cheating the general data/MC software, Gemc/Coatjava contracts!
  - e.g., `if(run==11)` or `event.has("MC::Particle")`, instead of the exact same code path
- Less config files, less necessary config parameters, less run-group specifics everywhere!

Note, all non-geometry CCDB constants were already by run number!

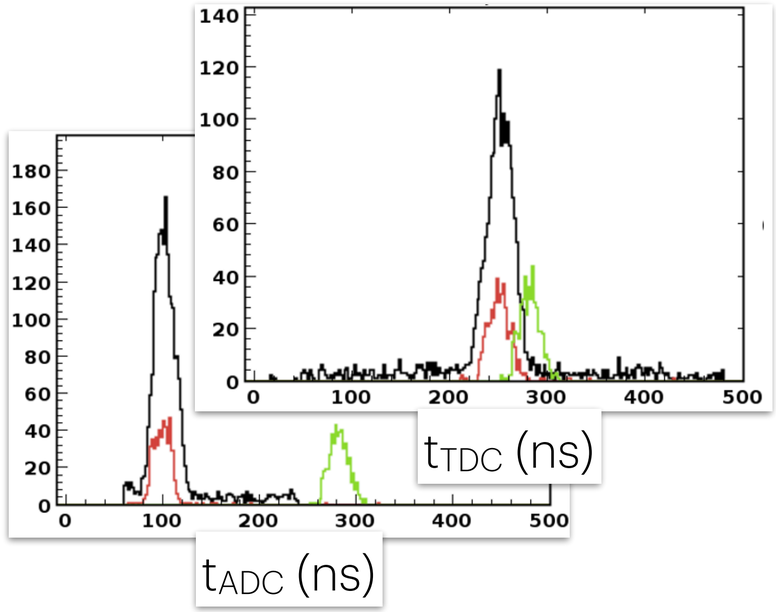
Real Run Number Simulations

Transition

- Backward compatibility
  - run=11 simulations still work as is, no changes!
- One new CCDB variation is currently required for real-run-number simulations
  - `mc_2026`, zero-misalignment trackers
- RRNS already available on the dev portal

What's Next?

- We can now better leverage our existing "run" databases, CCDB/RCDB/QADB
  - For example, sample detector conditions by run for simulations, map runs to integrated luminosity for sampling weights, and/or beam current for background merging, ...
- DC digitization overhaul has been in the plans
- Any detectors missing big run-dependent modeling affects?



- Gemc 6.0 and Coatjava 14.0.0 add RRN support, lots of uncheating on digitization stuff in both to make it possible
- CCDB updates were a large part of the job
  - map run-group variation to run-number range for geometry constants in the default variation

mcdj

https://pages.jlab.org/hallb/clas12/mcdj/

- A clas12 simulation pipeline runner, just to make running the simulation chain easy(er)!
- Intended for general collaboration use, motivated by testing needs, both manual and automated
- Advertised in weekly meetings

Examples

1. Run 12 parallel pipelines, 100 events each, using the dvcsgen generator and its t-min cut:

```
mcdj -j 12 -n 100 -g ./gemc.xml -y ./recon.yml -- dvcsgen -t 0.1
```

2. Process a bunch of LUND files, 3 concurrently:

```
mcdj -j 3 ... -- lund1.txt lund2.txt lund3.txt lund4.txt  ...
```

3. Use GEMC's particle gun:

```
mcdj -- gemc -BEAM_P="e-, 6*GeV, 15*deg, 20*deg"
```

4. Run 100 pipelines, 10 concurrently:

```
mcdj -j 10 -J 10 ...
```

5. Run 100 pipelines in series:

```
mcdj -J 100 ...
```

mcdj

mcdj runs the full CLAS12 simulation pipeline, starting from a clas12-mcgen-compliant generator, the GEMC particle gun, or LUND files. It also supports parallel and series pipelines, background-merging and truth-matching.

Requirements

- A reasonably modern python version.
- User-specified configuration files for GEMC and COATJAVA are required. See the optional clas12-config git submodule for standard setups.
- All executables must be in `$PATH`; mcdj does no envirnoment setup. The clas12 environment modules at JLab, and container-forge's simulation container, are examples.
  - clas12-mcgen-compliant generator, e.g., `clasdis`
  - `gemc`
  - `recon-util`
  - `bg-merger`
  - `dst-maker`

Installation

mcdj is just a python script. Download and run it.

Usage

Note, a double-dash `--` should be used to separate mcdj's options from the generator command-line arguments.

```
> mcdj -h

usage: mcdj [-h] [-n #] [-j #] [-J #] -g PATH -y PATH [-r RUN] [-m] [-s SEED]
            [-d] [-R] [-q] [-v] [-c] [-b PATH [PATH ...]] [--denoise]
            gen [gen ...]

positional arguments:
  gen                   generator command line or LUND file(s)

options:
  -h, --help            show this help message and exit
  -n, --nevents #       number of events per job (default=10)
  -j, --jobs #          number of parallel jobs (default=1)
  -J, --Jobs #          number of serial jobs (default=1)
  -g, --gcard PATH      GEMC gcard configuration file
  -y, --yaml PATH       COATJAVA yaml configuration file
  -r, --run RUN         run number (default=11)
  -m, --match           enable truth matching
  -s, --seed SEED       random number seed (default=clock)
  -d, --dst             run standalone dst-maker
  -R, --recon           disable reconstruction
  -q, --quiet           silence GEANT4 exceptions
  -v, --verbose         increase verbosity (repeatable)
  -c, --cleanup         delete intermediate outputs
  -b, --back PATH [PATH ...]
                        background files for merging
  --denoise             enable old denoising (use YAML for new)
```

NEXT