2x2 / Mx2 operations

Matt Kramer

DUNE general computing meeting
June 24 2024

BERKELEY LAB

DAQ

- Each subsystem (charge, light, MINERvA) has its own DAQ software with a dedicated machine underground:
  - Charge: crs_daq, built on larpix-control (Python)
  - Light: afi-daq (custom C++ DAQ from developers of readout HW) + Python control server
  - MINERvA: minervadaq (C++) + mnvruncontrol (Python)
- Combined run control software (MORCS) for coordinated run start/stop
  - Each DAQ responsible for starting/stopping subruns

Underground services

- Various DBs (PostgreSQL, InfluxDB, Prometheus) for slow controls, server monitoring, data quality monitoring
- Services that fill the above
- Grafana for the above
- Mostly in Podman containers
- noVNC remote desktops for things that aren’t web-based (Ignition, camera)

Data flow

- Charge and light data continuously copied to NERSC via rsync-in-a-loop
  - Enables low-latency prompt processing (nearline DQM plots, etc.)
  - These files are temporary; long-term replicas to be Rucio-managed
- All 3 systems able to produce MetaCat-compliant metadata
- xrootd server on charge DAQ machine recently tested (for ingest daemon)
- Plan is for all 3 data streams to go through MetaCat and Rucio
  - Replicas on dCache and at NERSC
- Tuning subrun lengths to ensure file sizes good for tape system

Nearline processing

- At NERSC, rsync’d charge/light files automatically run through series of prompt processing stages
  - Calibrated files for analysis
  - Nearline DQM plots (available on NERSC portal)
- MINERvA running nearline jobs underground

Runs / conditions DB

- Schema has been defined to enable matching of subruns across the 3 systems
- Also included: Slow control summaries, detector/DAQ configuration, etc.
- Currently sqlite. Eventually postgres (DUNE ConDB)
- Scripts aggregate data from slow controls, DAQs etc. to produce JSON blobs for upload to UConDB'
  - Currently read blobs off disk, fill sqlite
  - Eventually read them from UConDB, fill ConDB