Plans and readiness for 2x2 MC at NERSC

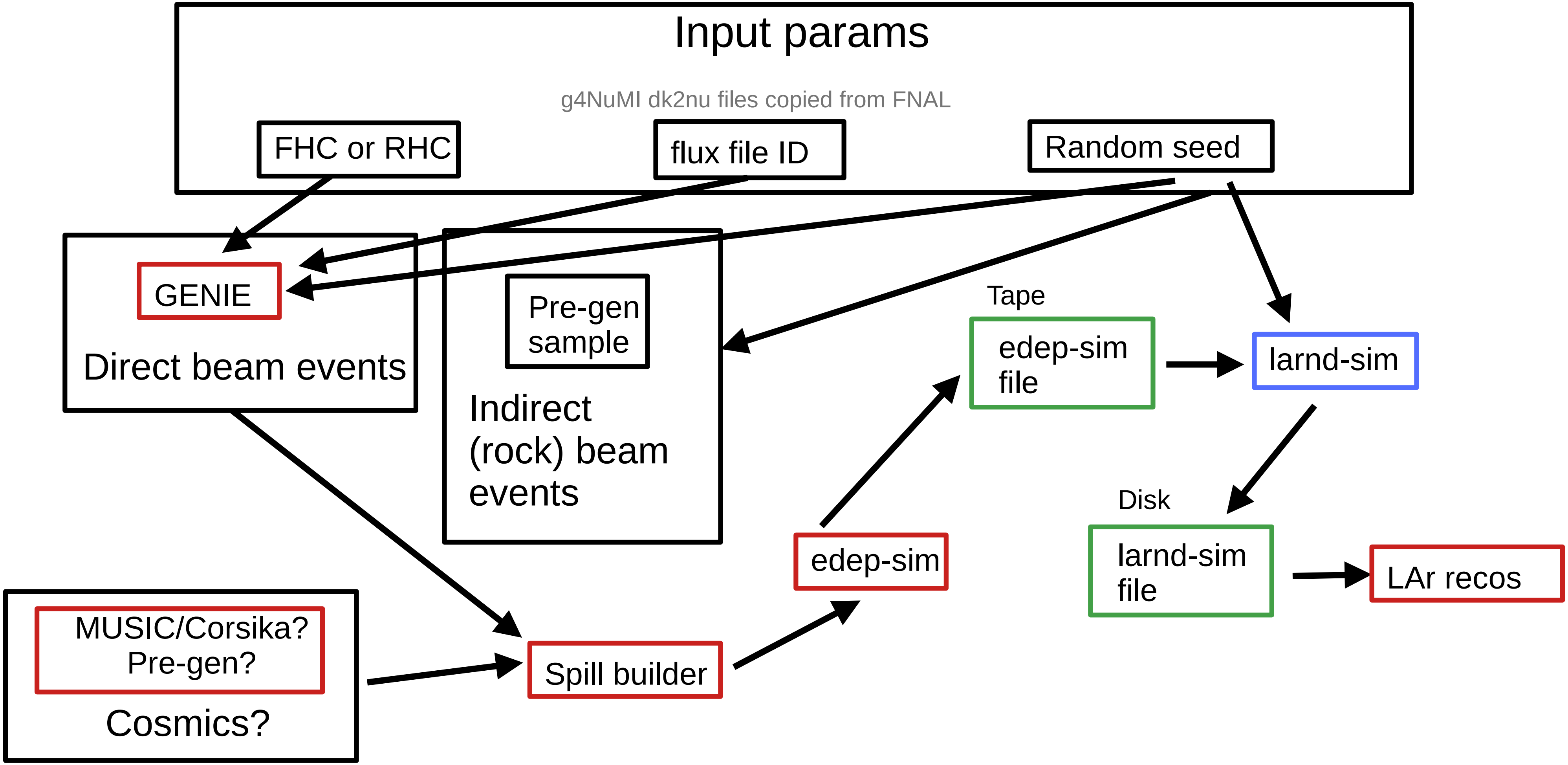
Matt Kramer, LBNL

2x2 analysis workshop, Bern
Jan 20 2023

Context

- **2x2 + MINERvA in NuMI beam soon**, expect ~5E20 POT (~1yr @ 50% uptime)
- **Need 10X MC statistics,** for both FHC and RHC
  - 20 * 5E20 = 1E22 POT simulated
- Goal is to quickly **produce this sample at NERSC** (Perlmutter GPU+CPU), make it available for analysis
- For both MC and data, also need **calibration and reconstruction** (flow, mlreco, Pandora)
- This talk is an update since ~~the last talk~~ the last talk

High-level workflow



Input params

g4NuMI dk2nu files copied from FNAL

FHC or RHC

flux file ID

Random seed

GENIE

Direct beam events

Pre-gen sample

Indirect (rock) beam events

Tape

edep-sim file

larnd-sim

Disk

larnd-sim file

LAr recos

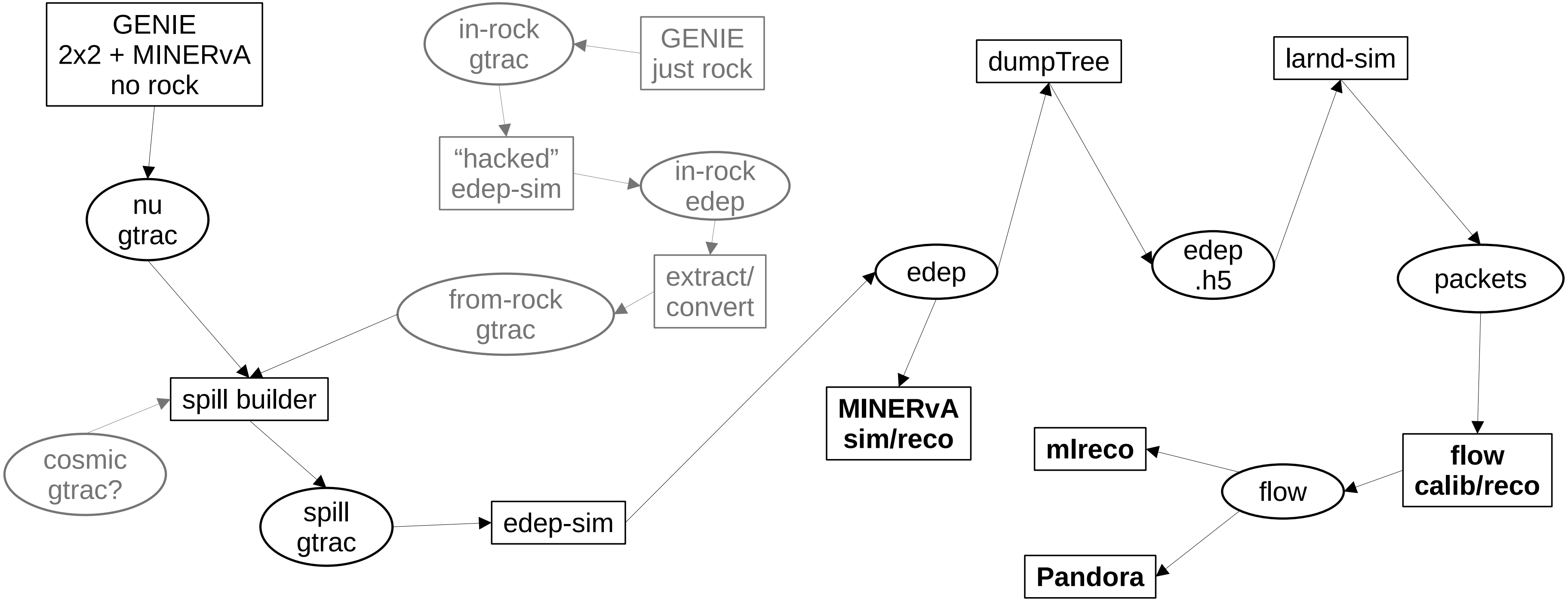
edep-sim

Spill builder

MUSIC/Corsika? Pre-gen?

Cosmics?

Detailed workflow



Rectangles = code, ellipses = data, gray = one-time, black = every spill

Production management: Fireworks

- Fireworks: Workflow software, widely used at NERSC
  - Python, YAML, MongoDB; flexible and easy to understand
- DB stores:
  - Configurations (code versions, environment vars, paths, ...)
  - Job parameters (FHC/RHC, flux file ID, random seed)
  - Records of completed (sub)jobs, failed jobs, etc.
- Worker (launched through Slurm) pulls items from DB, processes them. 6 subjobs => 6 workers
  - GENIE+edep-sim worker (CPU)
  - larnd-sim worker (GPU)
  - flow worker (MPI CPU)
  - mlreco worker (GPU)
  - Pandora worker (CPU)
  - MINERvA worker (CPU)

Data flow through workers (i.e. subjob dependencies) defined through directed acyclic graph (DAG)

Reproducibility

- Must be able to exactly reproduce simulated output based solely on job configuration in DB
- Configuration includes git commit hash of prod script repository
  - FireWorks worker verifies that the commit matches before launching scripts, aborts if mismatched
- All software packaged in containers
- Production scripts store the hashes of all required containers, verified at runtime
- Container-building scripts deterministic and under git control
  - Useful extra: Mapping of git commits to container hashes (potential headache-saver)

Future-proofing

- Desire to eventually harmonize production management with the tools used in the broader DUNE ecosystem
  - Precluded now by time constraints, urgency of this sample
  - Conversation started with ND sim/reco group
- Keep FireWorks-specific (Python) code to the bare minimum; as much as possible lives inside Bash scripts whose only “dependencies” are environment variables
  - Should make it “easy” to move the scripts to another production system (e.g. POMS, data-dispatcher)
- FireWorks DB data format is simple and intuitive; Mongo (JSON) → SQL mapping should be straightforward

Calibration and reconstruction

- **flow** to provide calibrated hits (for mlreco, Pandora...) as well as basic track-level reconstruction
- **mlreco**
  - Inference and training run without complaints on Perlmutter GPUs
  - Historically used edep-sim energy deposits
  - SLAC progressing in using LArPix hdf5 format, e.g. from larnd-sim
- **Pandora**
  - Uses CPUs, should work on Perlmutter, hasn’t been tested(?)
  - Recent progress from Pandora team in consuming larnd-sim output directly
- **MINERvA**
  - Gaudi-based sim/reco (CPU); Geant4 wrapper replaced with edep-sim reader (Nöe/Tammy)
  - Working on running at NERSC
- Will need “realtime” reco on data (at least flow) in order to e.g. implement certain DQ plots
- See Kevin’s talk on the “calibrated” data format (output by flow) for use by mlreco and Pandora

Rock events (“done”)

- Run GENIE on stripped-down geometry (just rock + empty hall)
- Run edep-sim on GENIE vertices, save all trajectories that enter hall
  - “Hacked” edep-sim to remove requirement of energy deposits in sensitive volumes
- Extract trajectories into “spoofed” GENIE file for use by spill builder
  - One “event” may contain trajectories originating from multiple points; cannot(?) store as a single GENIE event, instead store “blocks” of “subevents”

Cosmic events?

- Cosmic rate in cavern is ~2 Hz / m^2; comparable rate in 2x2
- Spill window is ~10 μs
- So, not something we need to worry about (right?)

Spill builder (in progress)

- Take expected # (per spill) of direct beam events, indirect (rock) beam events, cosmic events
- For each spill, Poisson fluctuate above to generate # of each
- Use spill structure to generate random times for beam events
- Use uniform distribution to generate times for cosmics
- Draw direct beam events sequentially from job's GENIE file
- Draw indirect beam events randomly from pregenerated sample
  - How big of a sample do we need?
  - Apply fluctuations to positions, directions, energies?
- Sort by times, merge into single GENIE file for use by edep-sim

Metadata, cataloging, replication

- Working on MetaCat installation at NERSC
  - “Exercise” by filling with singlecube run metadata
  - For simulation, copy job configuration+params from FireWorks DB
  - For 2x2 data, to contain run configuration, conditions, calibrations, etc. (replicated w/ FNAL?)
- Desirable: NERSC as a DUNE Rucio storage element
  - Enables automatic replication of MC from NERSC to FNAL, data from FNAL to NERSC
  - Automates tracking of locations on disk/tape

Projected compute/storage requirements

- What do we expect when including rock events?

Production timeline

- Full LAr chain (at least up to flow), including rock and cosmics: Aim to have ready mid-Feb
  - Main remaining action items: Finish spill building; cosmics; ensure reproducibility; MetaCat
  - mlreco, Pandora, MINERvA may take longer
- Once at that point, plan to generate a 1%-scale test sample (~1E20 POT)
  - Verify projections of compute/storage requirements
  - Run various analyses for validation (volunteers needed!)
- Once everything looks good, proceed at full scale