Fermilab

GENIE Experiences with Continuous Integration

Gabriel N. Perdue // Fermilab
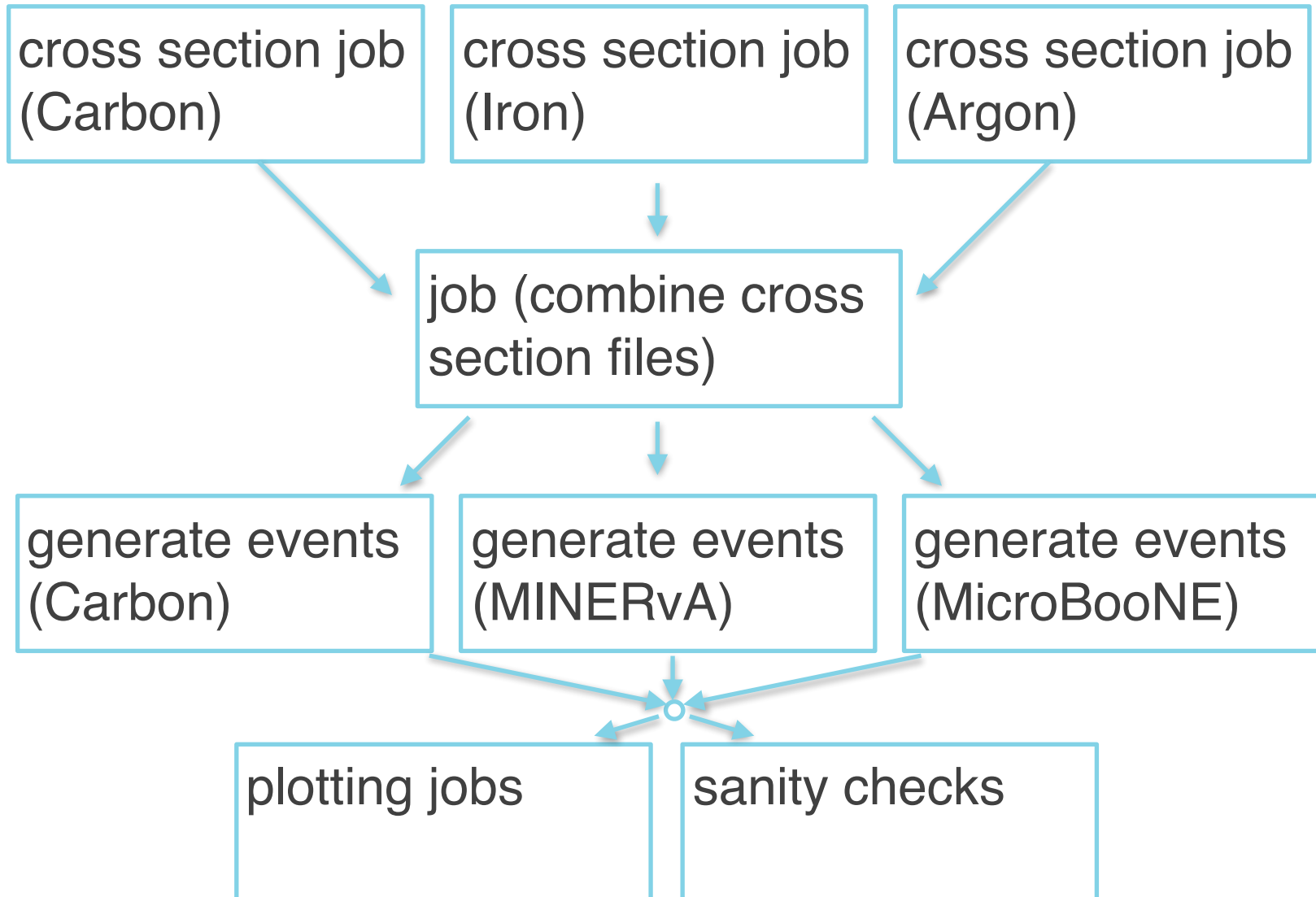FIFE Workshop 2017 // June 21st

GENIE and CI

- Very positive experience

- Our basic plan:
  - Nightly build of integration branch (don't need true *continuous* integration) with unit tests,
  - Weekly physics validation of integration branch (requires O(100) jobs on the grid and ~2 days (limited by slowest jobs in DAG) - currently using GPGrid but we are nominally OSG compliant),
  - Report from unit tests should go to Slack,
  - Link to report with plots, etc., from weekly validation should also go to Slack

- We got a lot of help with all of the above from the CI team.
  - Problems are on our end - grid scripting, etc.
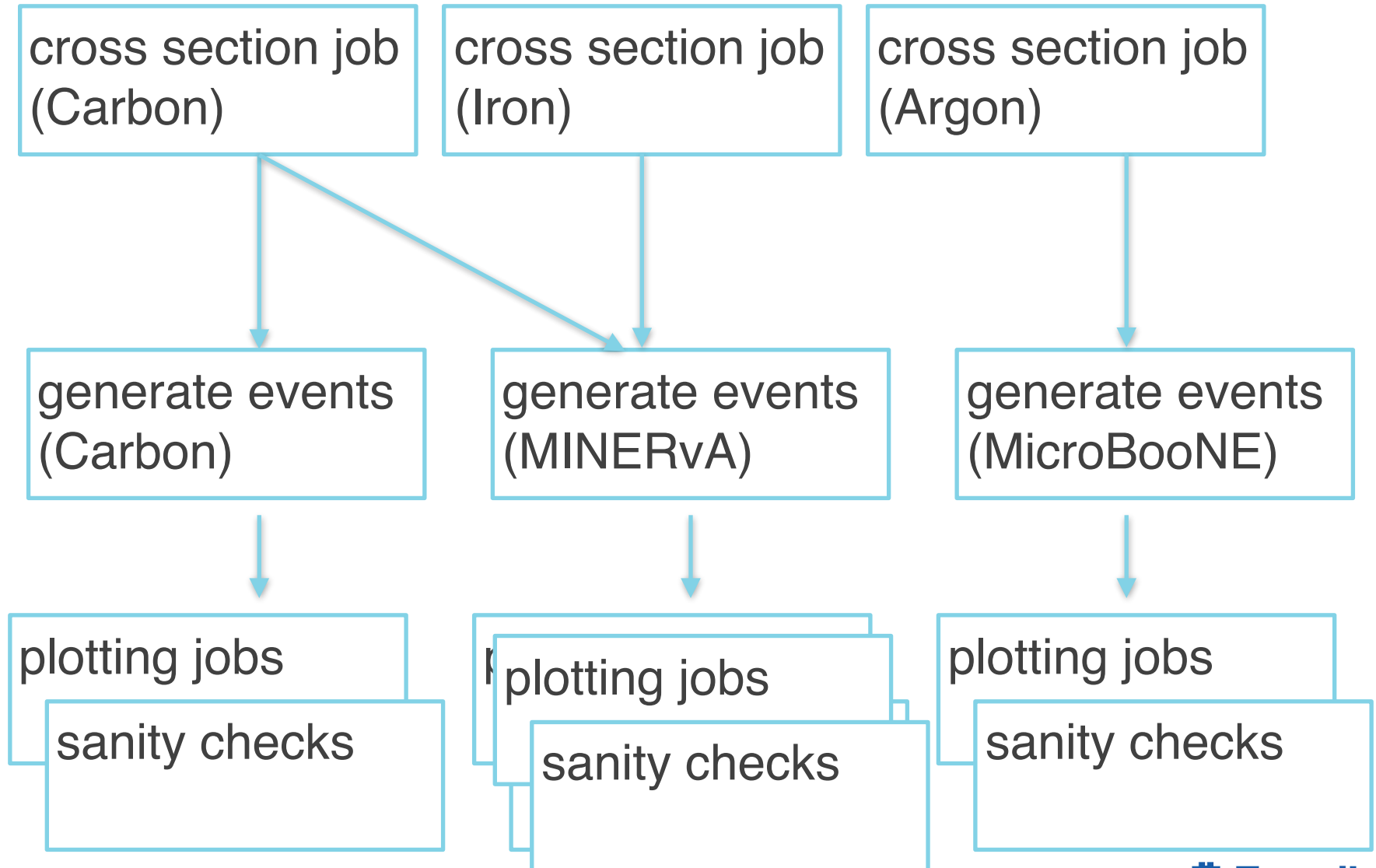   - Could always use help with that too…

Fermilab

Physics validation

```
┌─────────────────────┐  ┌─────────────────────┐  ┌─────────────────────┐
│ cross section job   │  │ cross section job   │  │ cross section job   │
│ (Carbon)            │  │ (Iron)              │  │ (Argon)             │
└─────────────────────┘  └─────────────────────┘  └─────────────────────┘
              │                    │                    │
              └────────────┐       │       ┌────────────┘
                           ▼       ▼       ▼
                    ┌─────────────────────┐
                    │ job (combine cross  │
                    │ section files)      │
                    └─────────────────────┘
              ┌────────────┼────────────┐
              ▼            ▼            ▼
┌─────────────────────┐ ┌─────────────────────┐ ┌─────────────────────┐
│ generate events     │ │ generate events     │ │ generate events     │
│ (Carbon)            │ │ (MINERvA)           │ │ (MicroBooNE)        │
└─────────────────────┘ └─────────────────────┘ └─────────────────────┘
              └────────────┐  │  ┌────────────┘
                         ○
              ┌────────┘   └────────┐
              ▼                     ▼
┌─────────────────────┐ ┌─────────────────────┐
│ plotting jobs       │ │ sanity checks       │
└─────────────────────┘ └─────────────────────┘
```

       춘 Fermilab

Physics validation (could be)

```
┌─────────────────────┐  ┌─────────────────────┐  ┌─────────────────────┐
│ cross section job   │  │ cross section job   │  │ cross section job   │
│ (Carbon)            │  │ (Iron)              │  │ (Argon)             │
└─────────────────────┘  └─────────────────────┘  └─────────────────────┘

┌─────────────────────┐  ┌─────────────────────┐  ┌─────────────────────┐
│ generate events     │  │ generate events     │  │ generate events     │
│ (Carbon)            │  │ (MINERvA)           │  │ (MicroBooNE)        │
└─────────────────────┘  └─────────────────────┘  └─────────────────────┘

┌─────────────────────┐  ┌─────────────────────┐  ┌─────────────────────┐
│ plotting jobs       │  │ plotting jobs       │  │ plotting jobs       │
│   ┌──────────────┐  │  │   ┌──────────────┐  │  │   ┌──────────────┐  │
│   │ sanity checks│  │  │   │ sanity checks│  │  │   │ sanity checks│  │
│   └──────────────┘  │  │   └──────────────┘  │  │   └──────────────┘  │
└─────────────────────┘  └─────────────────────┘  └─────────────────────┘
```

🔷 **Fermilab**

Scripting help

- Are these common patterns?
- Easy to imagine experiments and other groups wanting a workflow like this - simulation validation requires running the simulation.
- Any support for a common infrastructure / toolkit for this?
- So far, the largest obstacle to taking full advantage of all CI has to offer has been shifting infrastructure under our feet and lack of internal expertise/bandwidth for solving problems in grid failures, etc.
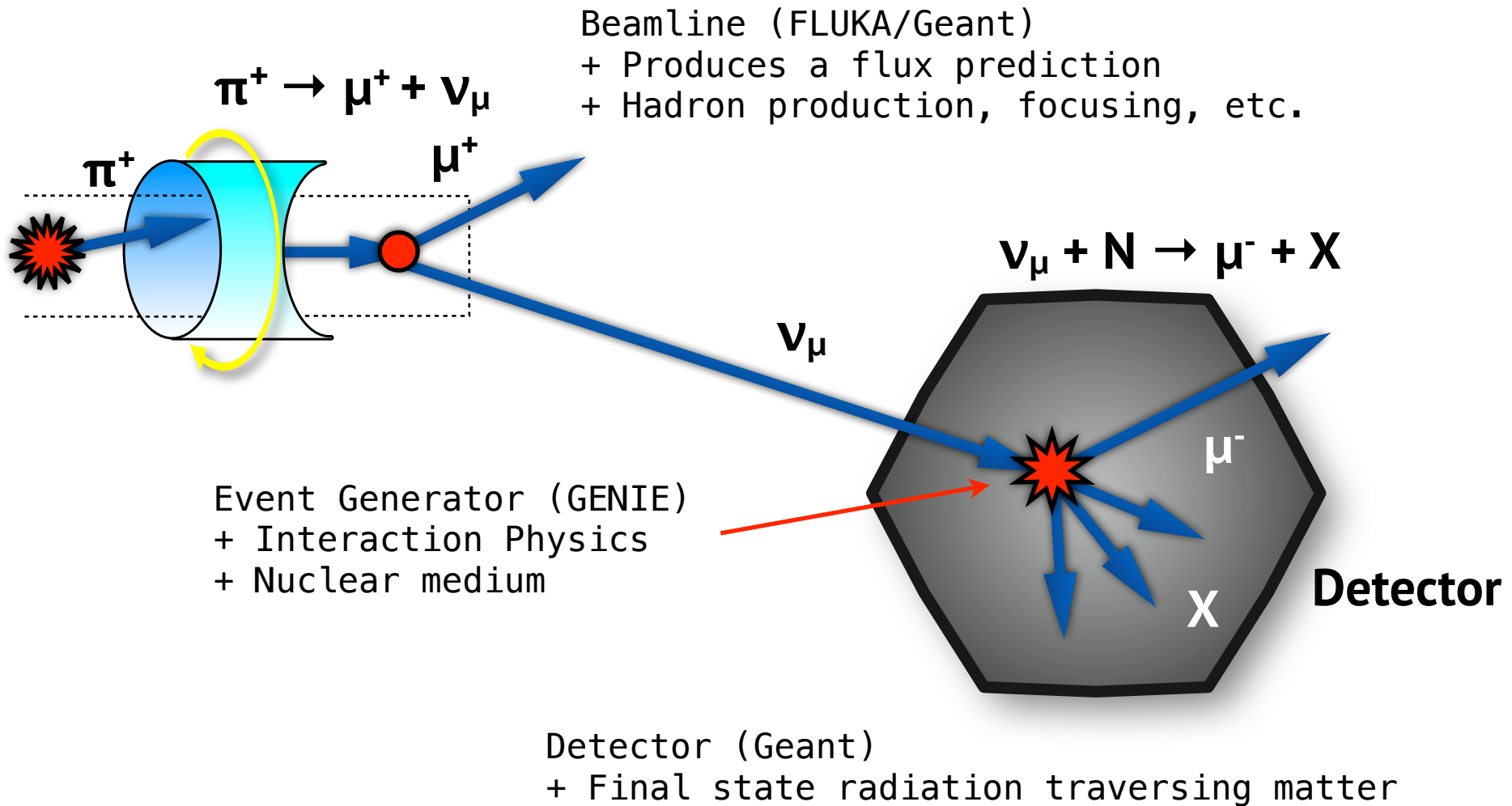
Fermilab

Other CI wish-list items?

- Nothing that is reasonable to ask for!
- We have a clear path forward for…
  - nightly (or even more frequent) builds and unit tests for the integration branch and other specific development branches,
  - weekly physics validation with performance performance and plots in pathways that are accessible to all of the collaboration,
  - ways to schedule special validation runs for release validation.
- We don't have experience actually *running* the weekly validation on a weekly basis through the CI and it is easy to anticipate some problems when we start, but we're confident that by working with the CI team we can make everything function smoothly.

🐝 Fermilab

Thanks!

🔷 **Fermilab**

Neutrino Simulations: A Three-Part Software Stack

Beamline (FLUKA/Geant)
+ Produces a flux prediction
+ Hadron production, focusing, etc.

$$\pi^+ \to \mu^+ + \nu_\mu$$

π^+

μ^+

$$\nu_\mu + N \to \mu^- + X$$

ν_μ

Event Generator (GENIE)
+ Interaction Physics
+ Nuclear medium

μ^-

X

Detector

Detector (Geant)
+ Final state radiation traversing matter

🔁 Fermilab

GENIE

- **G**enerates **E**vents for **N**eutrino **I**nteraction **E**xperiments.
- http://genie.hepforge.org
- Well-engineered C++ software framework built on sound OO-principles and design patterns. (The Gang of Four is omnipresent.)
- Propagates a flux of neutrinos (specified by function, histogram, or ntuple) through a geometry (Geant4-compatible) and simulates the initial interaction and propagation of hard vertex products through the nuclear medium. Geant4 takes over when particles leave the nucleus.
- ROOT provides many core utilities. GENIE also heavily leverages other HEP and FOS software - LHAPDF, GSL, Pythia, log4cpp, etc.

Andreopoulos, C. and Bell, A. and Bhattacharya, D. and Cavanna, F. and Dobson, J. and others. "The GENIE Neutrino Monte Carlo Generator". Nucl.Instrum.Meth. A614. 87-104. 2010.

춘 Fermilab

GENIE at FNAL

- GENIE is the primary event generator for:
  - ArgoNeut
  - SBND
  - DUNE
  - MicroBooNE
  - MINERvA
  - NOvA
- GENIE is being considered for special studies by MINOS and MiniBooNE (they use previous generation software for their main generators).

🐝 **Fermilab**