Ideas for GPU and portability in DUNE workflows

Brett Viren

November 13, 2026

Major ingredients / expertise for hardware/software portability

Simphony	optical physics and tracking on GPU, integrated into **phlex** framework via **edep-sim** LArTPC Geant4 app/lib.
Wire-Cell	C++, detector sim, signal processing, reconstruction, based on **TBB** flow graph.
TRED	Python PyTorch LArTPC 2D/3D drift and detector response sim, sparse block processing, dynamic batching.
SPNG	**Wire-Cell** signal processing, next generation, C++ **libtorch** base, fine-grained **TBB** multi-thread flow graph nodes running on configurable mix of CPU/GPU, tensors stay on device between nodes.
Phlex	already full ecosystem (“xerosere”) with ***WC**, **edep-sim**, **gegede** / GDML geometry, kinematics, raw **DUNE DAQ data**, **Arrow** data objects, **HDF5**. Phlex is based on TBB (synergy with WC) and support Python (TRED).
Build	portable binaries for full ecosystem via **Spack** (“multispack”), **pixi/conda** packages from Spack builds (“spaxi”)

Possible HEP-CCE portability thrusts

- Port Simphony to non-NVIDIA GPUs and/or high-core CPUs (ARM). Currently it depends on of NVIDIA OptiX/CUDA.
- Demonstrate and optimize full chain phlex/wire-cell GPU-heavy workflows on HPC.
- Port TRED Python / Wire-Cell CPU sim to SPNG-style C++ libtorch sim.
- Build and portable-binary deployment system for Phlex/xerosere ecosystem.