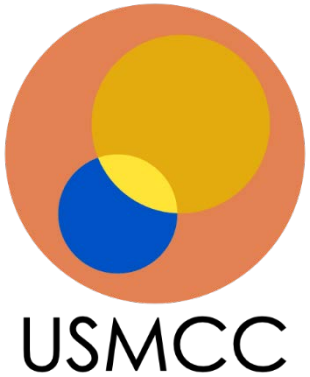
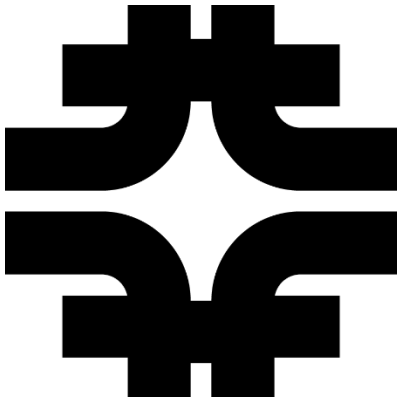
Computing Resources and Challenges

Kevin Pedro (FNAL)

August 8, 2025





Processing Chain

Generation

- $\mu^+\mu^-$ interactions
- beam-induced backgrounds

Simulation

- Particle interaction w/ passive & active detector elements

→ SimHits

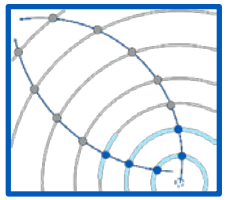
Overlay

- Add beam-induced background SimHits to $\mu^+\mu^-$ interaction events

Digitization

- Electronics response in each sub-detector

→ RecHits



Reconstruction

- High-level objects: tracks, vertices, jets, muons, electrons, etc.

Expected computational intensity of each step:

Generation	Simulation	Overlay	Digitization	Reconstruction
Inexpensive at LO; N^nLO can be much more intensive 🔎: negative weight reduction, GPU-based generators	FullSim is intensive; FastSim (Delphes) is cheap 🔎: GPU-based simulation; generative ML	Potentially most expensive step (BIB simulation in particular) 🔎: premixing, generative ML	Linear scaling w/ # hits 🔎: GPU porting?	~Quadratic (superlinear) scaling w/ # hits (classically) 🔎: Smart reduction, ~linear time ML clustering

Profiling

Step	CPU	Memory	Disk
BIB simulation	up to 24 hours/event (10^8 particles)	up to 32 GB/event (considering whole chain)	~20 GB/event (BIB)
BIB overlay	5 mins/event (before digitization!)		
Tracking	5 mins/event up to hours/event (depends on lattice)		~1 MB/event (signal, w/o BIB)

- These numbers consider the *current* simulation stack being used for design & physics studies

➢ BIB is main driver of computational needs

Available Resources

Major computing clusters:

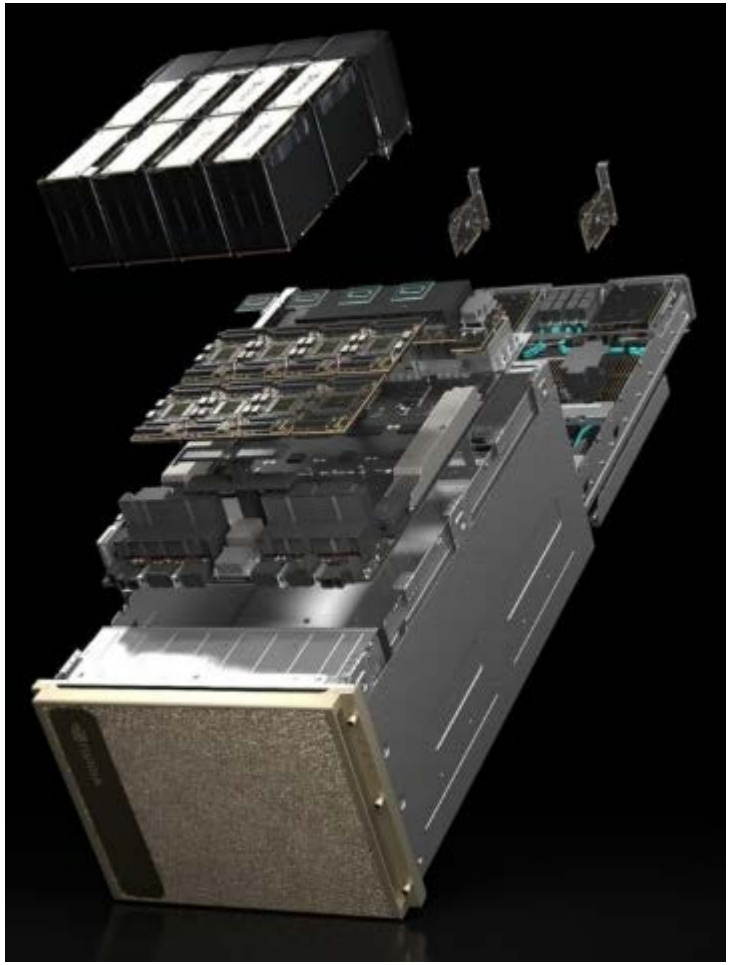
- lxplus (docs)
- DESY
- INFN
- OSG → dedicated!
- Fermilab LPC
- Analysis facilities (US, IT, DE, ES, …)

Future collider usage:

- Most major institutional clusters do not currently have dedicated resources
  - Batch CPUs available via user fair share as usual (with whatever memory they have)
- More difficult to find: disk space
  - Some at INFN, OSG

Heterogeneous Resources

- Most major clusters have *some* GPUs
  - Often partitioned or shared between users
- Different workflows/steps have different needs:
  - Code development: can live with partitioned/shared GPUs
  - Large-scale processing (training, simulation, etc.): need dedicated GPUs
    - e.g. from HPC centers
  - Analysis (e.g. ML inference): some analysis facilities provided specific inference servers (via Triton)
- Other alternative resources: ARM CPUs, FPGAs, etc.
  - Less widely available
  - Some providers have them, e.g. National Research Platform in US
  - Cloud: AWS (EC2) has F2 instances, GCP has TPUs, etc.



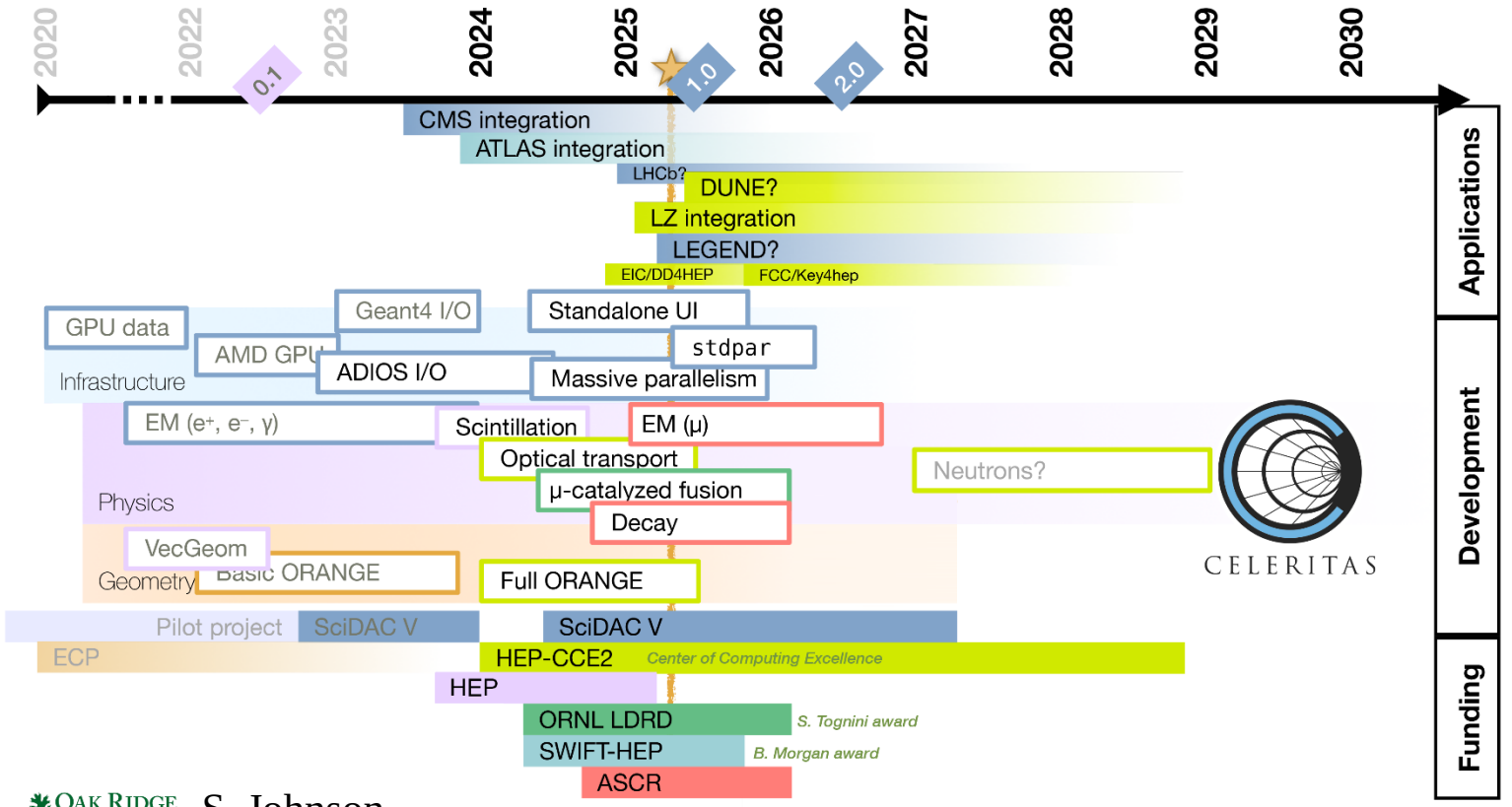
Data Management

- Experiments have operations funding to produce and manage data:
  - Data movement (availability, managing site storage pledges, etc.)
  - Metadata (provenance, versioning, physics info, etc.)
  - Discoverability (search, enumeration, access (tokens))
- Can there be a community-based, ground-up approach? Maybe!
  - Rucio: common tool now used by most experiments
    - Primarily for data movement
    - Also has metadata facilities
      - Avoid fragmentation of info across multiple databases
    - Users can upload custom datasets
  - Would need some central management, but could be mostly user-driven
- Muon collider data is complicated!
  - Many formats/products (FLUKA, geometry XML, ROOT, …)
  - Strong dependence on lattice (from BIB generation to tracking)

rucio.cern.ch

BIB Simulation

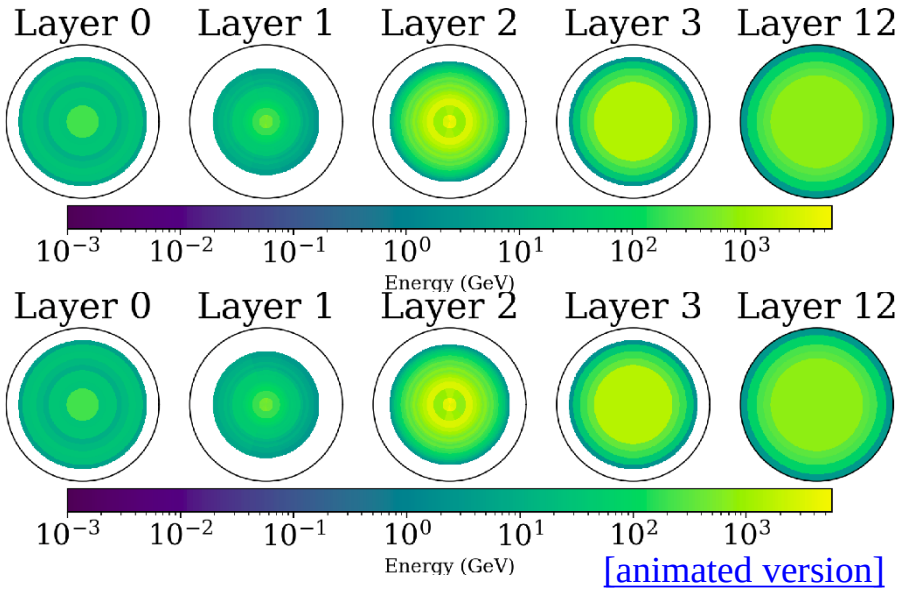
- Full simulation of 10^8 particles is necessarily slow
- How to speed it up:
  1. Run on GPU:
     - Exploit SIMD with huge batches (almost entirely photons and neutrons)



OAK RIDGE National Laboratory S. Johnson

BIB Simulation

- Full simulation of 10^8 particles is necessarily slow
- How to speed it up:
  2. Train generative ML algorithm:
     - GPU SIM hopefully provides sufficient events for training
     - Current ML4Sim efforts mostly condition on incident particle properties
       – BIB is a specific process: generate all particle hits together, condition on other relevant quantities (detector material/geometry/etc.)

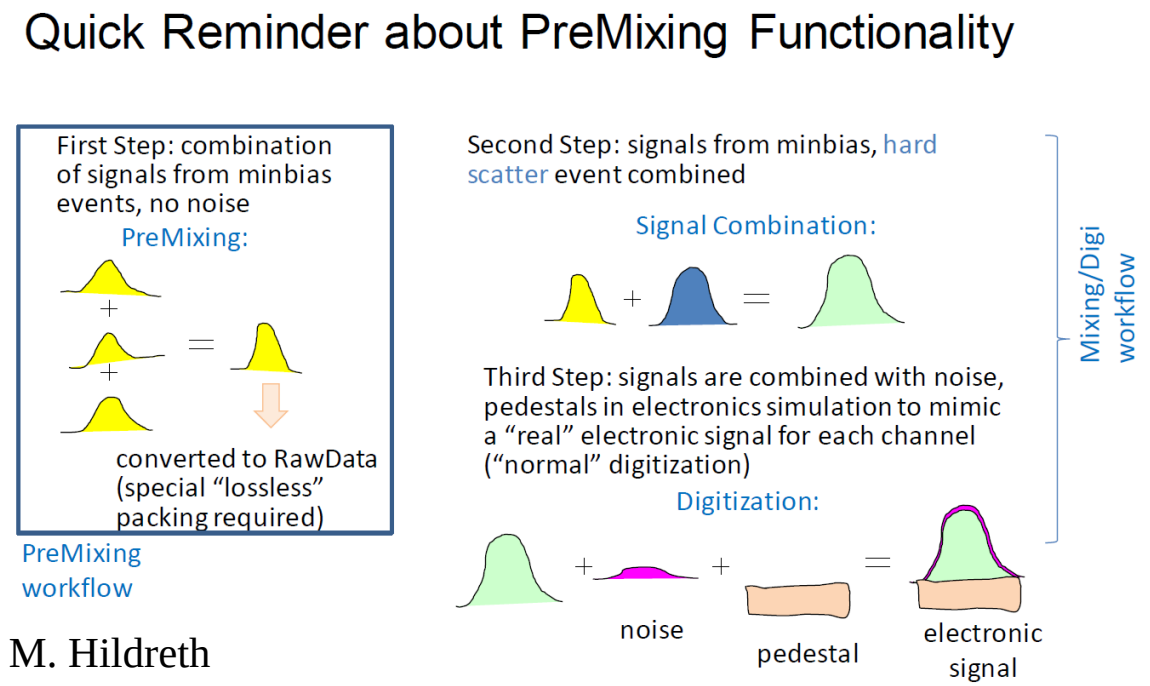


[animated version]

BIB Overlay

- Next step after simulating BIB; learn from LHC pileup overlay experience

1. Naïve approach: just overlay all simulated hits
   - Massively I/O intensive
2. Premixing:
   - Pros: amortize computing costs, compress hits
   - Cons: code maintenance (compression), I/O issues (large files, high availability), scenario-dependent (geometry, BIB profile, etc.)
3. ML-based:
   - Avoids both speed and I/O issues
   - Maybe generalizable to multiple scenarios?



M. Hildreth

Conclusions

- Muon collider has unique computing challenges
  - Can learn from LHC for some aspects
    - Both what to do and what not to do
  - Other aspects quite different
    - e.g. “on-the-fly” pileup mixing, currently being explored for CMS, not feasible for BIB (10^8 particles)
- Data management is important for reproducibility
- Particular challenge: develop a tightly-integrated design loop between accelerator and detector
  - With few dedicated computing resources: no running experiments yet!
  - Aim to be creative and try to grow our resources over time
- Need to support each other within the community in order to succeed