ECCE Computing

Cameron Dean
Los Alamos National Laboratory

08/04/2021

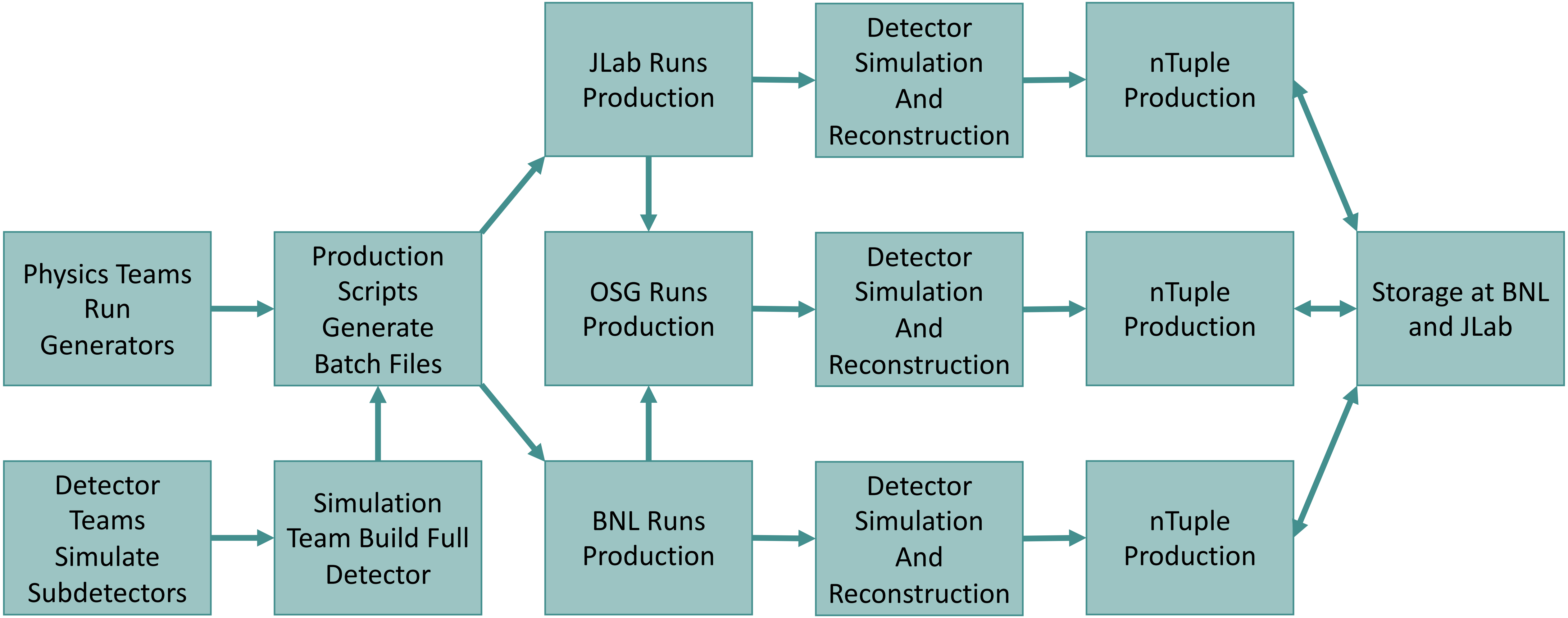
EICUG Summer Meeting

Overview

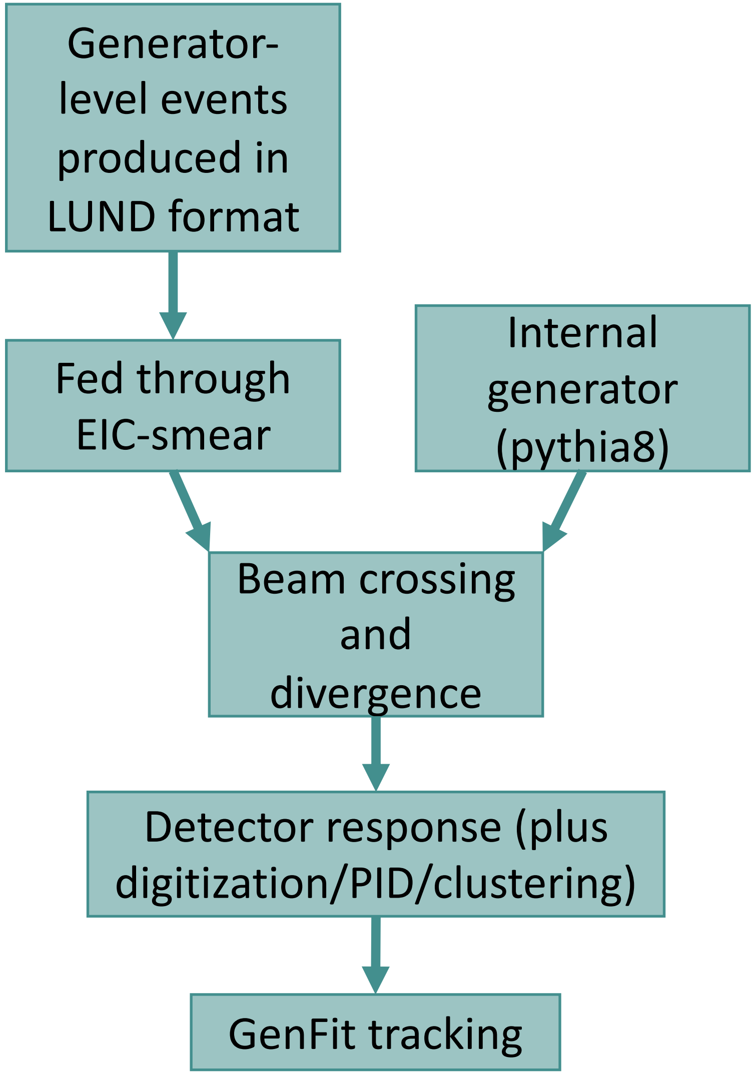
- ECCE computing is entangled in several areas of our consortium
  - Simulations with the detector groups
  - Reconstruction and nTuple production for physics working groups
  - Data production and storage
  - User access and communication
- Also using AI/ML to help optimize detector design and physics potential
- ECCE has progressed quickly to obtain a realistic detector simulation
- Physics teams have identified key processes to study this year
- Simulations and Computing teams have produced large data sets to meet these requirements

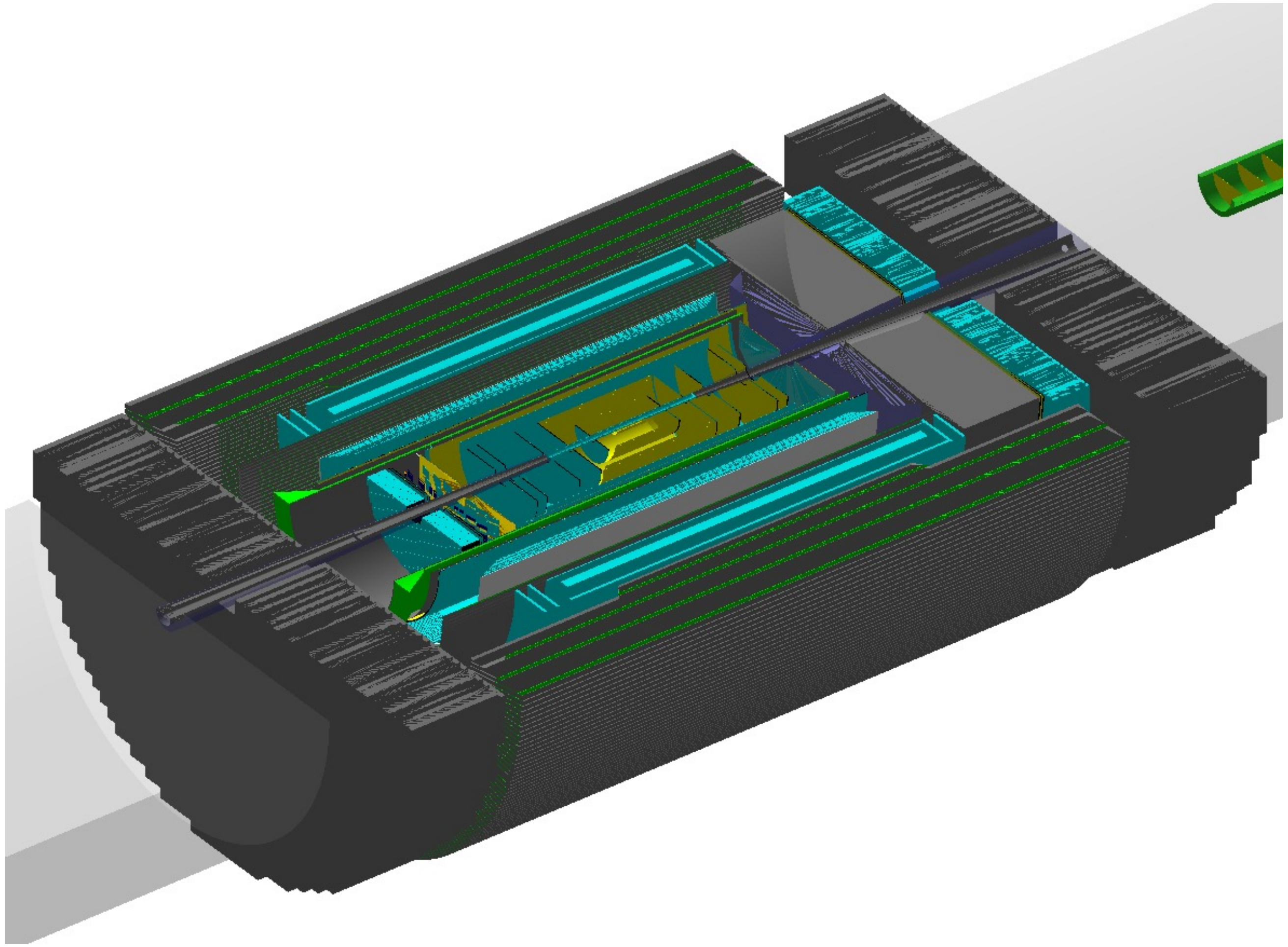
ECCE overview talk with more information

Production Workflow



Detector Simulation & Reconstruction

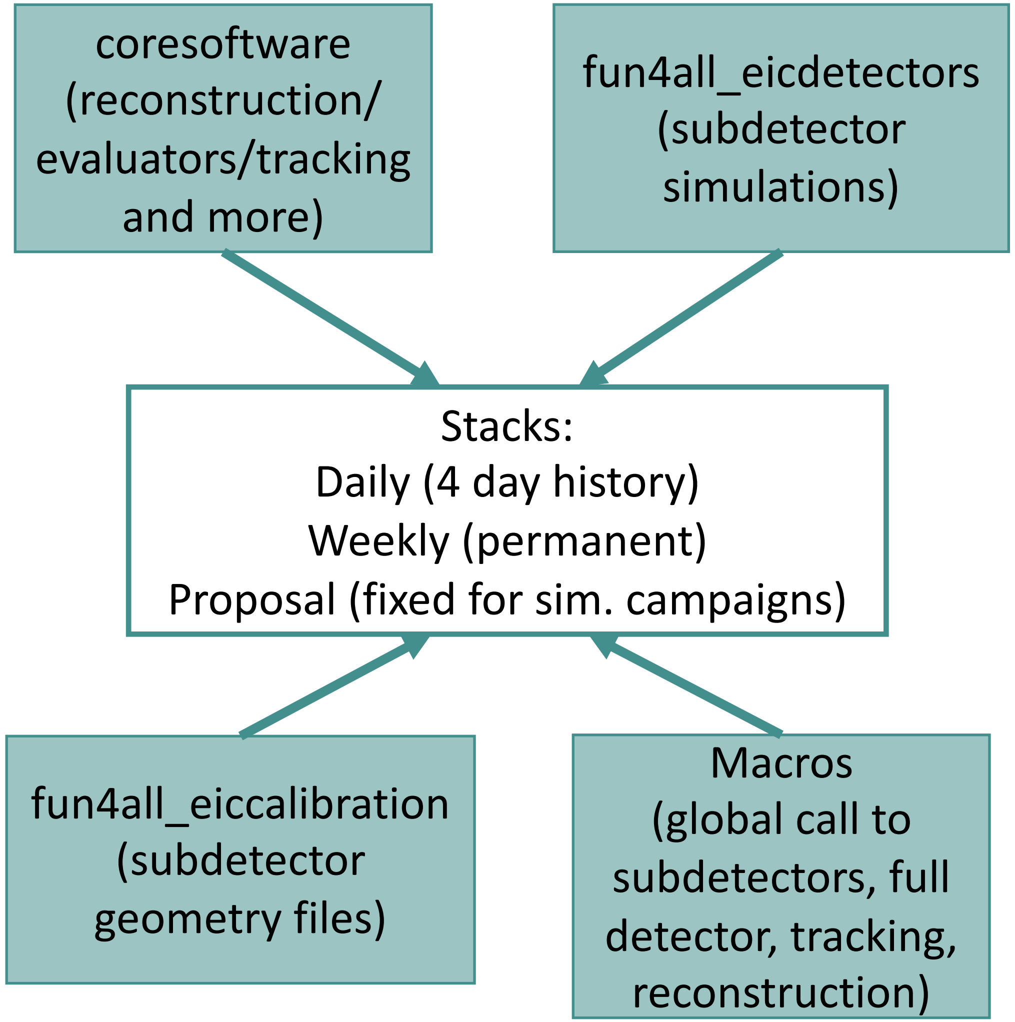




ECCE detector talks with more information:
Calorimetry, tracking, PID, far forward/backward

Code Production

- All users are welcome to submit packages or updates
- Analysis code is written in C++
  - Other languages are used for more specific tasks (e.g. python and bash for productions)
- Code-conventions established by Chris Pinkenburg
- All code must compile with clang, have no unused objects, cpp-check must have no serious issues, valgrind used to find memory leaks



ECCE software talk with more information

(Re-)Producing Physics Simulations

- Each collision sample (generator and beam conditions) produces a dataset file and associated ROOT nTuples (direct analysis)
  - DST has truth info., hits, clusters, tracks, jet reco. and more
  - Typical file is 1k → 5k events
- Reproducibility is very important
  - Requires 3 pieces of information: software stack for reconstruction, subdetector geometry and more, detector simulation file for global design and seed.
  - First two are easy with archived builds and git tags
  - Third one can require user intervention to set seeds to reproduce the data
- Our solution: Use RooUnblindPar
  - 3D random seed generation from input file name, number of events in file and starting point of input reader
  - Unique for each DST and reproducible anywhere with no user interaction!

Metadata logs

- Each production job automatically writes metadata to help debug issues

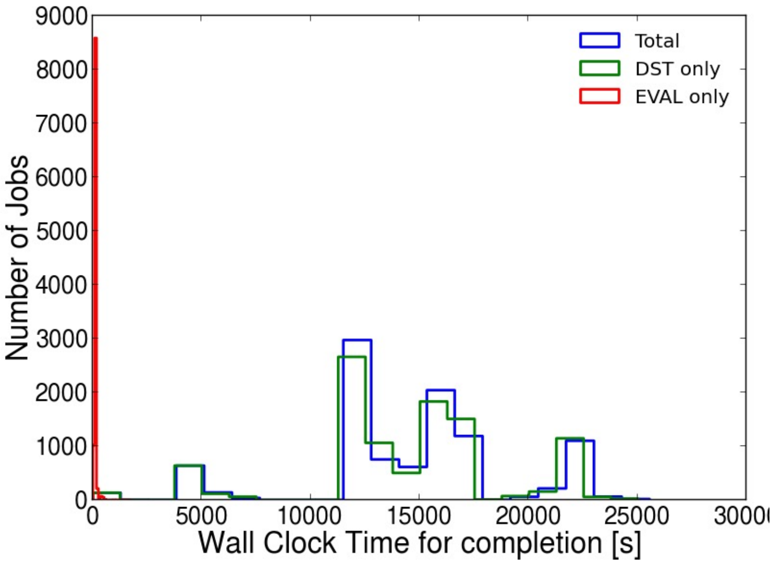
```
====== Your production details ======
Production started: 2021/07/25 17:10
Production site: BNL
Production Host: spool0680.sdcc.bnl.gov
ECCE build: prop.2
ECCE macros branch: production
ECCE macros hash: c131177
PWG: SIDIS
Generator: pythia6
Collision type: ep-10x100
Input file: /gpfs02/eic/DATA/YR_SIDIS/ep_10x100/ep_noradcor.10x100_run001.root
Output file: DST_SIDIS_pythia6_ep-10x100_000_0000000_05000.root
Output dir: /gpfs/mnt/gpfs02/eic/DATA/ECCE_Productions/MC/prop.2/c131177/SIDIS/pythia6/ep-10x100
Number of events: 5000
Skip: 0
=====================================
Seeds:
1322570549 (plus more)
md5sum:
01da8efd4555739dfa18fd96ee5b6a36
```

Fully and uniquely defines seed

Production Statistics and Notes

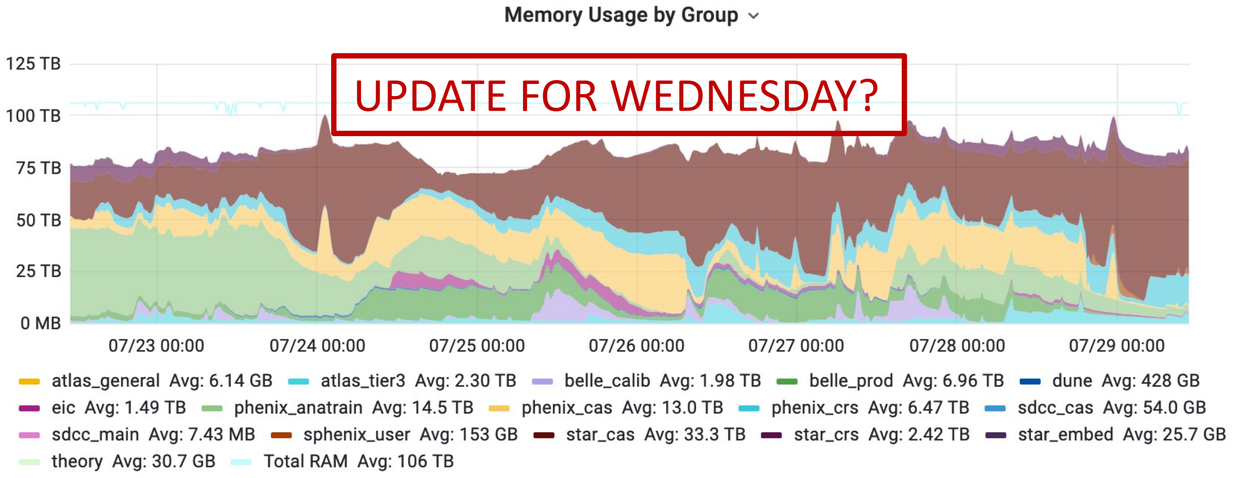
Planned simulation campaigns	2
Predicted events per campaign	120M - 160M
Typical event size	200kB
Typical event generation time	7s
Total storage per campaign*	30 TB
Typical job memory size	< 1.5 GB
Current events simulated in campaign one	101.3M

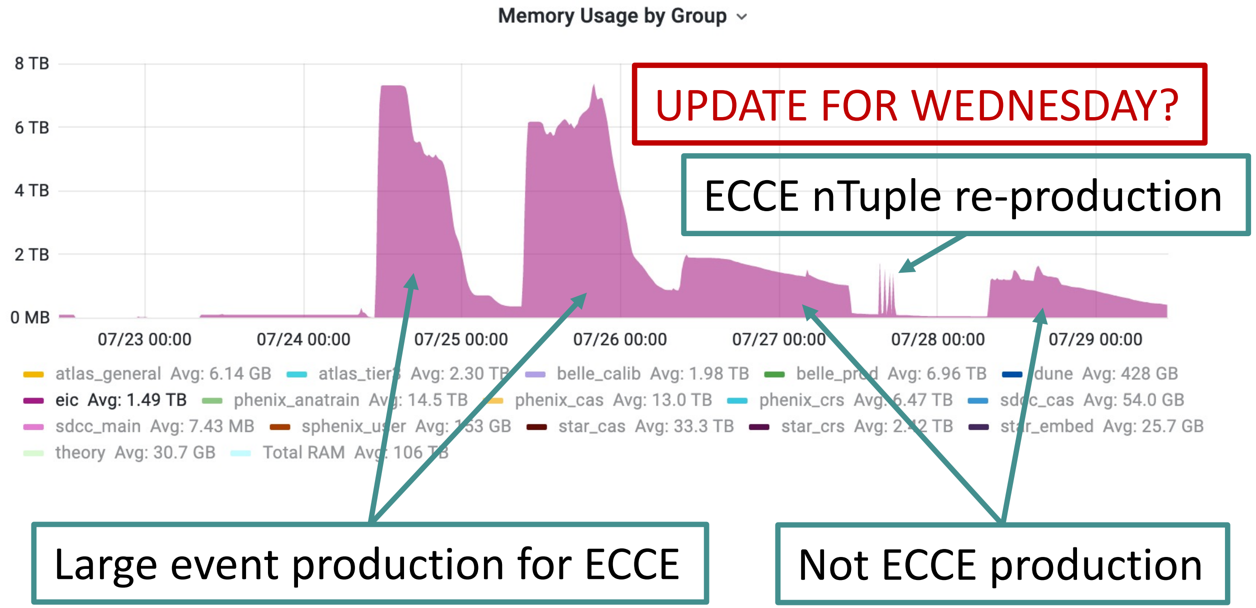
*We have not decided if we want to keep the first campaign raw data when we run the second campaign

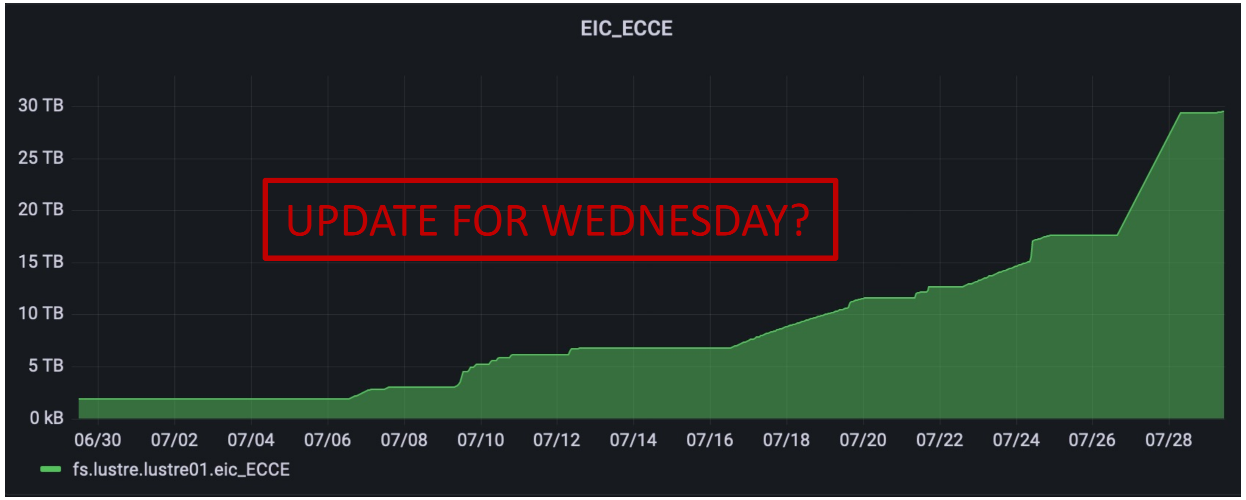


- First simulation campaign is under way, "June 2021 Concept"
- Second detector design is being finalized today, "July 2021 Concept"
- Production sites show > 90% job success rate (OSG > 98% success rate)
- Decided to separate simulation and reconstruction from nTuple production (former is time/resource intensive, latter is in current development)
- nTuple production automatically produces revisions and runs quickly (~2.5ms/event)
- To-do: add job monitoring and resubmissions (in beta-testing)

BNL Usage



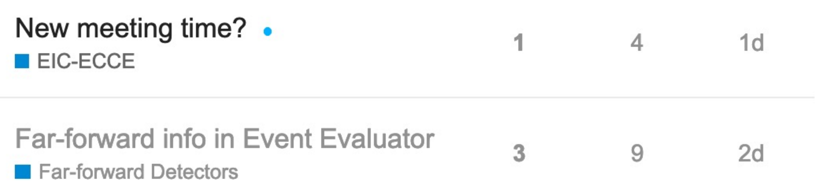




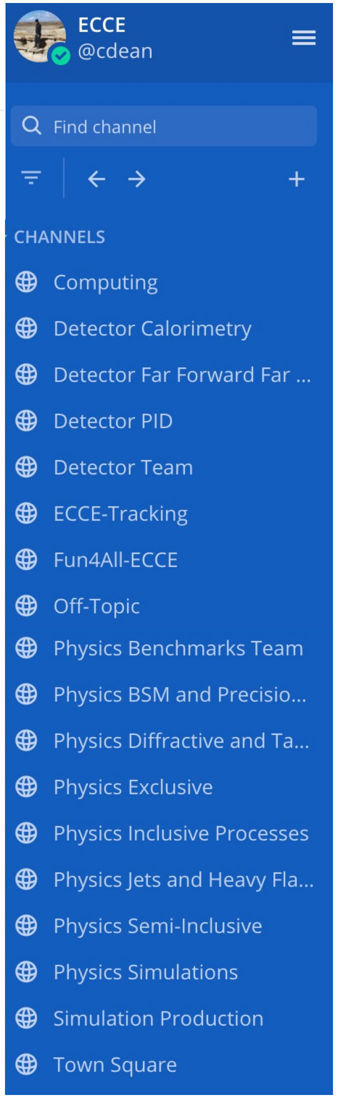
- EIC Condor use is relatively small but noticeable
  - Note, there is no distinction between ATHENA, CORE and ECCE on condor
- S3 storage is creeping up for ECCE
- ~140M events in storage so far

User Work Methods

- Many users across the world, not everyone joins with active BNL membership
  - Not feasible to get everyone accounts in proposal timescale
- Singularity (and VirtualBox) is used to distribute daily software stacks (and simulations)
- S3 (BNL) and xrootd (JLab) protocols used to distribute data
  - minIO client and read-access keys are distributed from ECCE stack
- Users are encouraged to use low-volume nTuples over DSTs
  - Keeps bandwidth to a minimum
  - Many physics plots can be made without large data processing

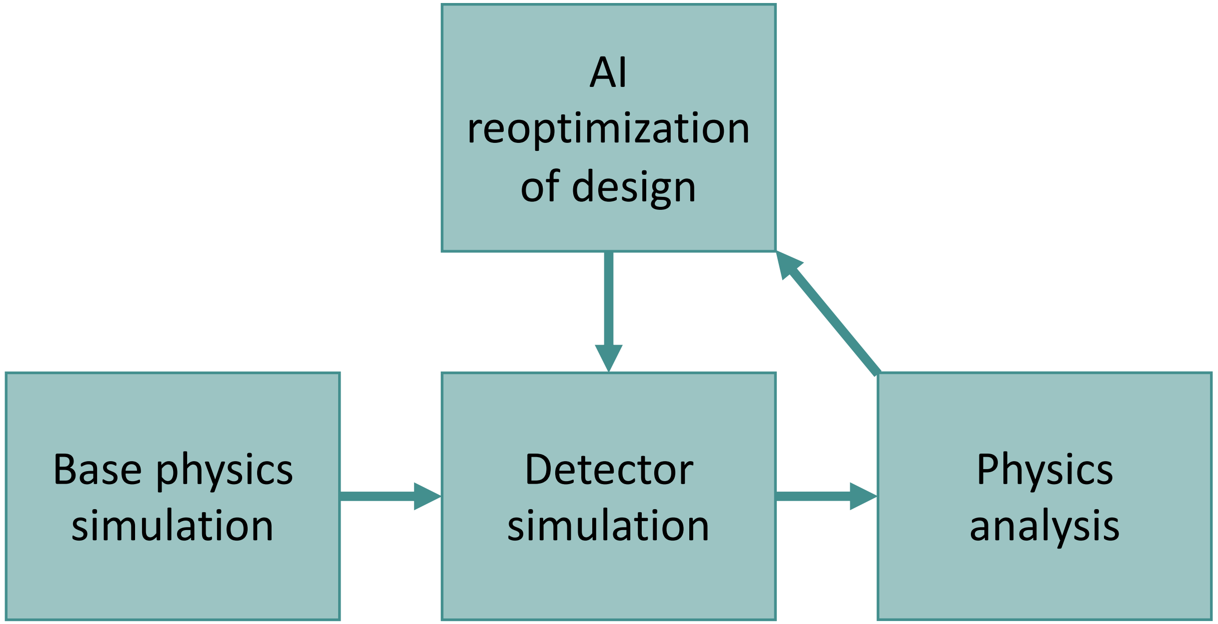


We use mattermost and discourse for quick communication. We can manually authenticate, no need to have a BNL account



Co-design at ECCE

- What is co-design of particle detectors?
  - Using machine learning to optimize the layout, material and performance of detectors
- Detector design is not a 1D problem: physics goals may be met with reduced materials or new technology may be needed to reach targets



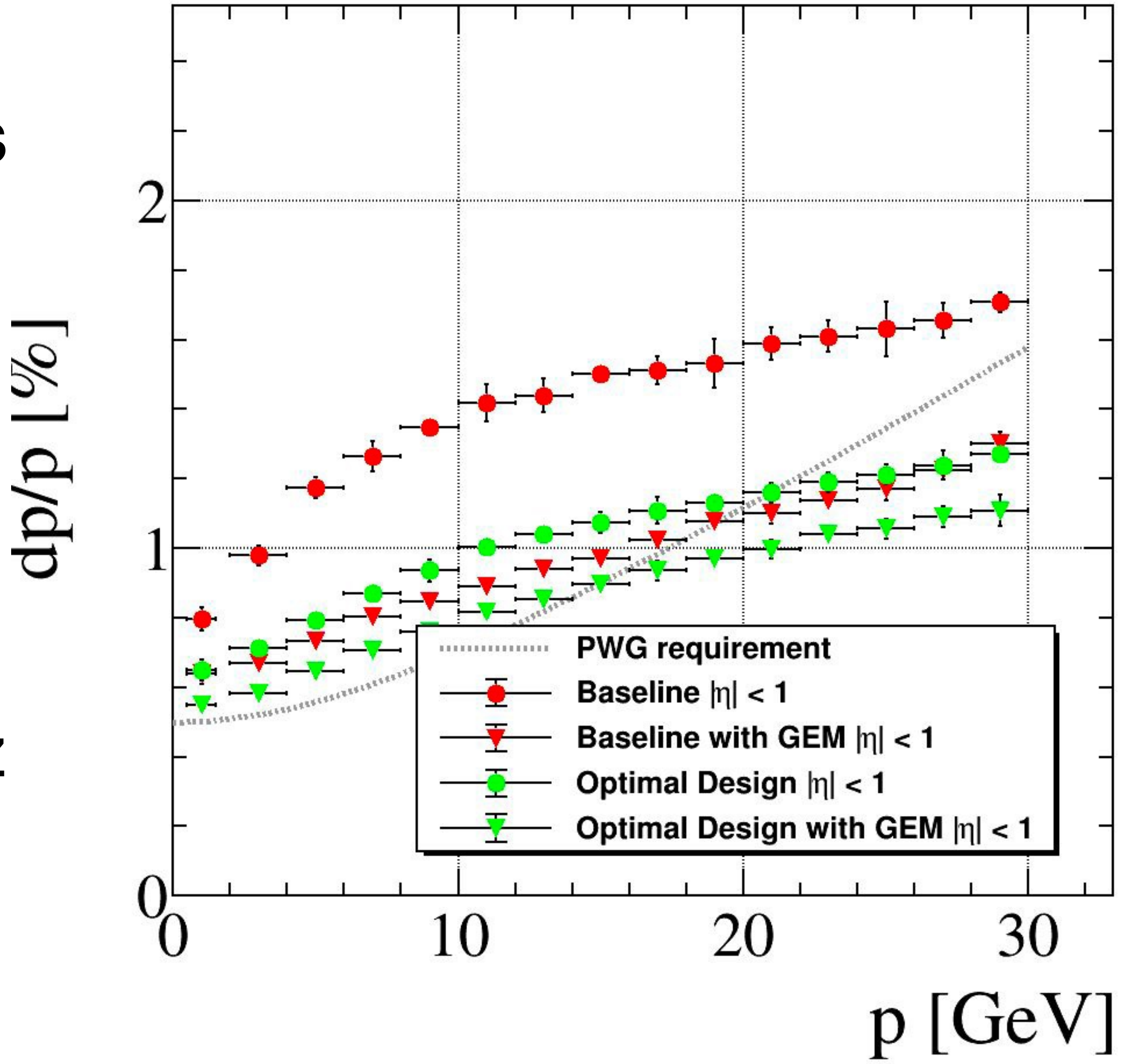
Tomorrows talk on co-design with more information

Co-design example: Tracking

- Using AI to simultaneously optimize barrel and forward/backward trackers
- Optimise for Kalman filter efficiency, DCA, momentum and angular resolutions
- Constraints:

1. Outer barrel radius < 51 cm
2. Vertex layer radius < 15 cm
3. Furthest disk position < 125 cm in z

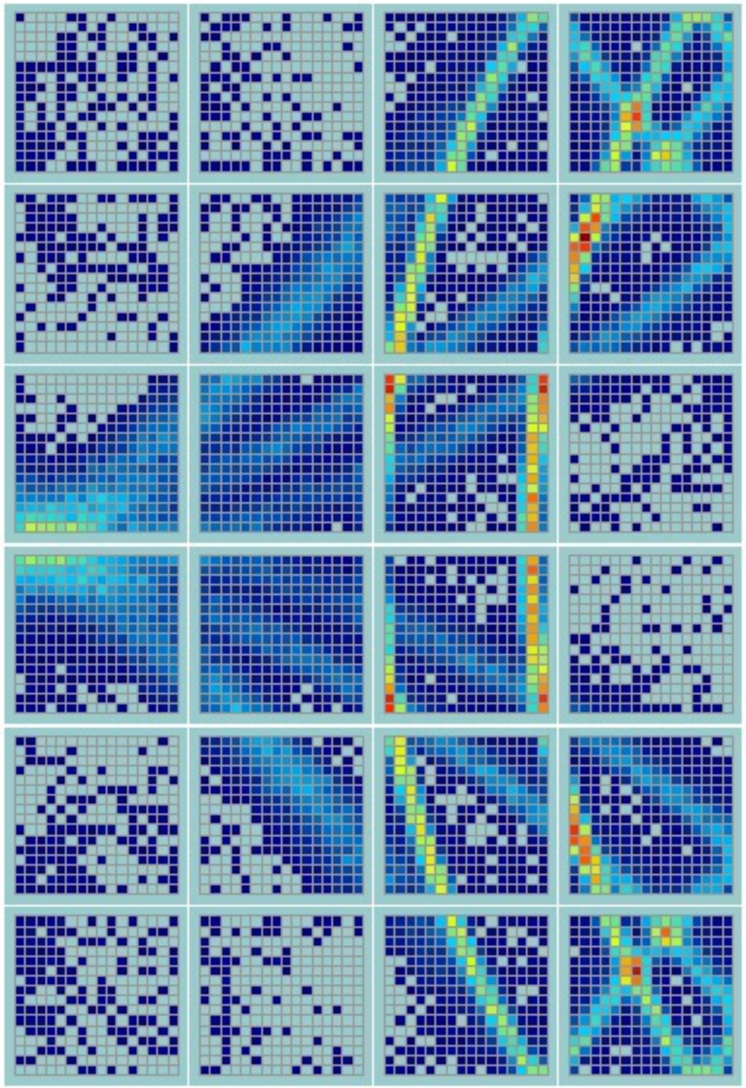
- Appears that optimizing placement gives larger change than additional detectors



C. Fanelli and K. Suresh

Co-design example: PID

- Optimisation of hpDIRC:
1. 5 configurations for focusing system
2. 4 settings for readout
3. 4 configurations for expansion volume
- Next steps are to optimize:
1. Bar width
2. Prism depth
3. Lens radius
- Optimisation performed using scikit-learn
- Right – $6\ \mathrm{GeV/c}\ \pi^{\pm}$ at 30^{o}

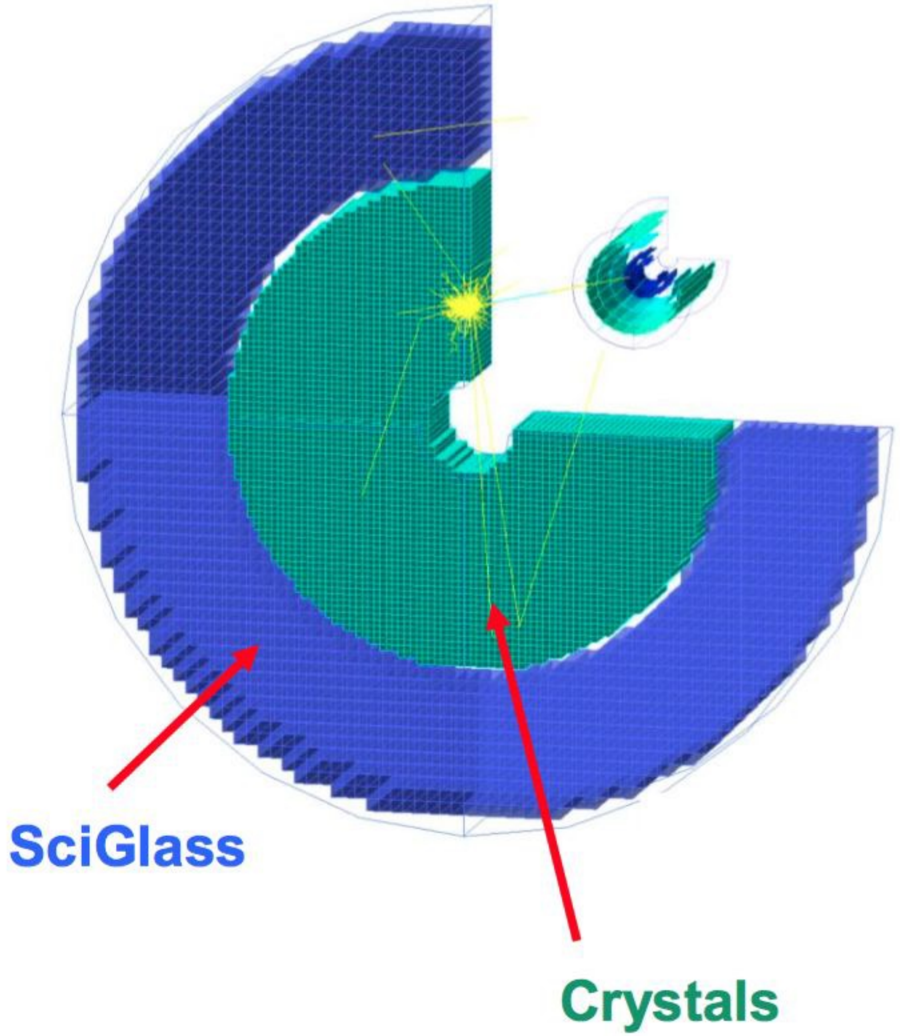


G. Kalicy, W. Phelps, A. Quiroga and N. Wickramaarachchi

Co-design example: Calorimetry

- Work has started on design on electron-going endcap EMCal
- The problem: Can we maintain resolution while reducing the number of crystals?
- Optimises:

1. PbWO crystal geometry
2. Inner/outer calo. radius
3. Densities
4. Efficiencies



V. Berdnikov, M. Bondi, C. Fanelli, Y. Furletova, T. Horn, I. Larin, D. Romanov, R. Trotta

Conclusions

- ECCE members have risen to the challenge of the detector charge
- Realistic detector simulations are in place
- Large data samples have been produced to understand capabilities
- Computing resources are used to ease user experiences, communication and data access
- AI working group is looking to the future with continual optimization

Thank you

Backup