What have been the two most significant accomplishments of Year 1 so far?

- **PAW -**
  - Started the miniapp packaging with Spack (p2r) and containerization (FCS), including improving the functionalities of FCS and implementing CI/CD workflows
  - Made progress in two representative workflows (Ported DUNE 2x2 NDLAr workflow to Polaris, ATLAS workflow on Perlmutter for integrating Globus Compute)
- **SIM**
  - Completed Geant4 -> Surface model translation
  - Cerenkov, Scintillation optical photon generation models integrated

What are the focus areas of the work in your technical area right now?

- **PAW -**
  - Mini apps
  - Workflows
- **SIM**
  - Optical photon stepping loop with absorption and Rayleigh scattering
  - Safety distance and geometry tracking optimization

What are the two or three main priorities for the next 18 months (till the end of Year 2)? Do they match our stakeholders' priorities/requirements/use cases? How much effort would it take to deliver on these priorities? What other CCE areas should work with you on these?

- **PAW**
  - Finish the packaging of p2r and FCS; Include WCT and patatrack in the next miniapp development.
  - Work out how to run ATLAS workflows with Globus Compute on Perlmutter, and then try to port to a different system (Polaris, for example)
  - Finish the DUNE workflow porting, testing and benchmarking on Polaris, and then replicate on Frontier
  - Cookbook based on PPS finding
  - Yes, they all match our stakeholders' priorities (HPC is important to them, but executing complex workflows on HPC is hard)
- **SIM**
  - Complete and validated optical photon transport in Celeritas + Geant4 integration
  - Full platform portable geometric tracking on ATLAS, CMS, and LZ models
  - Demonstration of optical photon simulation on LZ models
  - These match stakeholder priorities (LZ, Calvision)

What other CCE areas should work with you on these?

- **PAW**
  - SOP?
    - Metadata is important/ maybe this is an area we can work with SOP on (start with a survey on what/how metadata are being captured across HEP experiments)

    - SML?
        - We may want to use some ML workflows as our use cases
        - Conversely, maybe we can learn from the SML group what they have found
- **SIM**
    - PAW
        - Evaluate platform-portability models
        - Integration with testbeds and CI
    - SML
        - Training for fast and low-order models

Based on what you have learned about the IRI/HPDF program from Debbie and Lavanya's talks, are any of CCE's priorities aligned with IRI's goals? How could we benefit from IRI work? Could CCE work help IRI progress? Any suggestions for the Task Force idea?

- **PAW**
    - We can provide input to IRI/HPDF based on our experience with the workflows
    - IRI could help standardize access to different compute facilities (software environments, authentication/authorization, automation tools)
    - CCE should play the role of HEP advocate to HPC facilities, and help IRI consider HEP's needs; our experience and the workflow portability layers can contribute to IRI's software considerations.
    - **IRI/HPDF Task Force:** Should have both CCE stakeholders and CCE members involved; HEP-CCE understands the commonalities of HEP compute challenges.
    - Maybe HEP-CCE can learn from the experience on the NSF HPC systems
        - Outbound connectivity
        - Wants to support HEP
        - Cvmfs availability

How would you use AI/ML methods to achieve your goals? Do you have in-house AI/ML expertise, or would you need to collaborate with some AI/ML specialists outside your area?

- **SIM**
    - We would need to collaborate with AI/ML expertise that largely exists outside of the current SIM workforce
    - AI could be used for image feature detection for geometry validation
    - LLM methods could be used to accelerate user workflows, e.g., auto-generation of problem inputs
- **PAW**
    - Could use LLMs to write a cookbook-style AI assistant for parallelization
    - Or do automatic code optimization/parallelization
    - Generate documentation with LLMs

    - Or can we develop a HEP-specific LLM?
    - Need to have a robust unit testing/CI/CD framework for verification and validation
    - no in-house expertise in CCE, but no doubt at our labs or larger groups
    - Validation / Verification - ML model can point you to which variables you need to look at after code has been modified

Based on what we know about the FASST program (https://www.energy.gov/fasst, Rick Steven's talk), can HEP-CCE contribute data, methods, or applications for HEP foundation models?

- **SIM**
    - Simulation could be used to provide training data for surrogate low-order models, parameter optimization, and detector software trigger design
- **PAW**
    - contribute experience running complex workflows
    - HEP-aware AI assistant for HPC software