Phenomenological interpretation of (cosmological) data

- Example: extract cosmological parameter constraints using survey data such as catalog of galaxy clusters (~10^4) with gravitational weak lensing profiles
  - Some recent examples from my group
    - Clusters+3x2pt combined probes: https://arxiv.org/pdf/2412.07765
    - ETHOS n=0 Dark Matter constraints: https://arxiv.org/pdf/2411.19911
    - Testing Modified Gravity: https://arxiv.org/pdf/2409.13556
  - Fits generically many cross-experiment or multi-probe analyses
    - Ease of executation would be attractive to significant community of users
  - Input and output data in this example are of moderate size, so „cloudy computing“ model matches well
  - A single „production run“– analysis process focused on adding and validating elements to the modeling, meeting (blinded) tests and then a final run is unblinded and published

Use Case Workflow

What happens now…

- Create refined input dataset that can support the analysis
  - Selection crucial, iterative with testing and decision making
- Prelocate input dataset on target cluster
- Install needed software on target node
- Prepare Slurm job and submit
  - Multi-threading and MPI enabled
- Monitor job (to track convergence of posteriors)
- Collect final results, extract parameter posteriors
- Evaluate success of (blinded) run, plan changes to job to address inconsistencies, and resubmit until happy

The PUNCH experience

- Archive interface for finding science ready data
  - Interfacing to existing data collections, making new collections available
- Data placed in S4P
- No containers- make software available using CVMFS?
- Submit job to C4P
  - Target node matching needs of job selected (including HPC), job submitted, job runs and streams input data from S4P
  - Output data streamed to S4P
- User evaluates results, alters job appropriately and resubmits
- User captures DRP from final run

Responses Thomas‘ questions (1 of 2)

- This Use Case connects to all three pillars and builds upon existing PUNCH-1.0 tools
- Which problems?
  - Broadly applicable use case
  - Cloudy computing OK
- Which gaps?
  1. Integrating archive „find and analyze“ for data
  2. C4P use without containers
  3. Jobs deployed on HPC centers
  4. Simplifies „single production run“ jobs and enables DRP production

- Generalizable:
  - Archive “data find and analyze“ utilities broadly applicable in Astro/Cosmo– could be more broadly useful
  - A container-less thread for C4P may make our services more attractive and has broad applicability
  - Integrating HPC allows for a PUNCH operations model that can be built in partnership with the major public computing centers
  - More compute flexibility helpful everywhere. DRP production to capture efforts also an attraction

Responses Thomas‘ questions (2 of 2)

- End-to-end FAIR use case?
  - Archive „find and analyze“ function supports broad access to forefront datasets
  - Dependence on the analysis– could also support analysis of proprietary data.
  - Published results typically require publication of data and algorithm: „FAIRness“
- View outside PUNCH
  - „typical“ use case that emphasizes simplicity and flexibility– if it works as envisioned it then allows outside users to deploy their use cases on PUNCH without strictly adopting all HEP-like „best practices“
- Operation model:
  - PUNCH tools working on HPC resources then offers a model where PUNCH team and user base partner with HPC centers for the needed resources
  - Software deployed on CVMFS would quickly converge to the „core tools“ needed by the communities, and there would be little/no software overhead
    - No containers required for „single production run“ analyses
    - DRP may still require containerizing the production run that produced publishable results
- Sustainability connected to user attractiveness of PUNCH services