HEP-CCE Questions to the Experiments:

Questions about your experiment's computing:

1) Have you updated your resource projections for the next decade?
Have you identified any "pain points" that require significant R&D?

Our resource projections for the next decade are still vague. We will be able to refine them when the first data is in hand. Compared to the major HEP experiments (ATLAS, CMS, DUNE), our needs are probably small.

2) What is the role of HPC systems in your experiment today? Are you planning to increase this role in the future? Do you envision significant changes in your computing strategy that do not involve HPC?

We are currently using NERSC as our major computing facility. With data arrival, our needs will very likely increase.

We have experimented with high-throughput (grid) computing but it was not particularly successful for our workflows.

3) If this particular issue is a concern, what is your strategy for increasing heterogeneity (ARM vs. x86, NVIDIA vs. AMD, FPGAs, ML accelerators)?

We have a diverse set of tools that we have to run. Some of them are developed by the Rubin Project (the data processing tools and some QA tools) and they will be running on x86. Some of our simulation tools can take advantage of GPUs, given our focus on NERSC, we have targeted NVIDIA.

4) What are the experiment priorities for algorithmic R&D (e.g., pattern recognition, simulation, data management, and analysis pipelines), and what are the associated strategies (e.g., physics optimization, parallelization, ML, dedicated resources)?

- Bespoke cosmological simulations: generation of simulated galaxy catalogs with systematics (bias, intrinsic alignment, baryonic effects) and choices of cosmological parameters tailored to needs of working groups
    - Strategies: improvements to physics models, GPU parallelization
- Efficient running of Rubin Science Pipelines and management of results: "reprocessing" of fraction of Rubin data with variant algorithms, additional data, etc.
    - Strategies: Application of workflow management tools
- Optimization of analysis pipelines - MCMC chains for science analysis and data pipelines for measurement
    - Strategies: Machine learning, optimization of hardware I/O and processing

5) What are the experiment strategies for I/O and storage optimization from physics-driven data reduction to lossy compression and network and storage-driven workflow optimization?

So far, we have not encountered major I/O issues, this might change with data arrival, e.g., running the Rubin code, we do see some contention for disk access as early parts of their processing pipeline are I/O intensive. If we need to run those parts at significant scale at NERSC, there may be some need for optimization, but it would have to be done jointly with, and perhaps primarily by, the Rubin project.

6) What else is important to know about your experiment computing needs?

Questions about current and future HEP-CCE work:

1) Are you familiar with the CCE evaluation of portability layers for parallel applications? If so, are the layers we are working on (Kokkos, SYCL, alpaka, OpenMP/OpenACC, std::par) the right ones to evaluate? Did we forget something?

We are not familiar with this evaluation, but would be interested in the results. You might also wish to consider OneAPI.

2) Are you familiar with HEP-CCE's I/O and storage (IOS) activities? These include (1) an experiment-agnostic mini-app that mimics HEP application I/O patterns, (2) measuring and understanding I/O patterns of HEP applications with Darshan, and (3) an HPC-friendly HDF5 mechanism to write HEP data originally serialized using ROOT I/O. Are these relevant to your I/O issues? Can you suggest other directions that will be useful for you?

Cosmology data access patterns are usually somewhat different to typical HEP patterns, and most of the methods described in this question do not sound particularly applicable. Profiling using Darshan might be valuable, if it works on NERSC's Perlmutter machines' file system.

3) The HEP-CCE IOS effort started two study groups dedicated to (1) HEP event data models that are efficient for both disk I/O and accelerator offloading and (2) "intelligent" data compression (domain-specific, guaranteed precision, and recoverable precision). Are these directions relevant to your experiment R&D plans? Are there other I/O projects that CCE should focus on?

These also do not sound applicable to DESC's mission. Incidentally, domain specific data compression has long been in use for compression of astronomical images, and there may be prior art your study group would find useful.

4) There are R&D activities in the HPC and HEP communities on Object Stores (e.g., HEPnOS, RNTuple/DAOS). HEP-CCE IOS has relevant expertise in this area. Would you be interested in an “event object store” (or equivalent analog) R&D activity?

The Rubin pipelines make use of object storage in order to provide access to their pipeline data products both from cloud and on-prem resources. Similarly, DESC's analysis pipelines produce a variety of data products, and we anticipate our pipelines will be run on various resources in different locations, so object storage solutions in those contexts would be useful.

5) The CCE complex workflows group is exploring the applicability of Parsl/FuncX to HEP workflows. FuncX is already used as a resource abstraction layer by Coffea, and ATLAS is testing it as an endpoint for their complex ML workflows. Are you familiar with this work? Could your experiment profit from a FaaS or a complete workflow execution engine that abstracts away the resources (clouds, grids, HPCs) it is running on?

A FaaS layer would in principle be useful for DESC since we have several different computing centers where we could run our pipelines, and those centers have different kinds of resources available.

6) HEP experiments use HPCs mainly for high-throughput, embarrassingly parallel workloads but are exploring more complex workflows for use cases like adaptive learning and simulation-based inference. As part of a workflow management system (e.g., a backend for FuncX), HPC runtime systems provide advanced capabilities such as task/resource-level resilience, maximizing task execution concurrency, dynamic resource managing and partitioning, and online performance profiling. How important is it to enable those capabilities in your experiments?

All of those capabilities would be useful for DESC computing, and dynamic management of resources is a particularly significant problem for which we don't have a reasonable approach. Especially for running the image processing pipelines, the resource needs of the tasks evolve significantly such that a batch allocation that would be appropriate for the early parts of the pipeline would be poorly configured for later parts.

10) Any other inputs about CCE's current work and future plans?