By allowing to run using an empty I/O workflow specification and monitoring I/O accesses for one cycle, we can propose an I/O description to the user to simplify the specification.

4.3. Smarter I/O Scheduling

From the list of opportunities, we realise data placement and migration in a heterogeneous (multi-storage) environment. These goals will be achieved via the proposed I/O-aware scheduler, called here EIOS (ESDM I/O Scheduler). EIOS will make the schedule considering Cylc workflow and ESDM provided system characteristics. We are working together with Cylc Team in developing how EIOS interfaces with Cylc. While Cylc schedules the workflow, EIOS can provide hints about colocating tasks which provide the opportunity for keeping data in local storage. Our design imposes only minor changes to Cylc as the core requirements are covered by normal functionalities:

The ability to dynamically set the job (Slurm) directives for a task

> This will be achieved by calling an external command (run on the Cylc scheduler host) which adds additional directives to be used by the job. This command, provided by EIOS, will determine some attributes of previous tasks through simple SQL queries to the Cylc database. We plan on using the Cylc broadcast functionality to change the directives used by a task by running an external program prior to any task where we need to be able to alter the directives.

The ability to dynamically set storage locations

> This will be achieved by defining environment variables in the job script which are set to the output of another external command (run on the job host). This command, also provided by EIOS, will have access to all the normal Cylc environment variables with details about the current task.

We plan on utilising DDN's IME API to pin data in IME and to trigger migrations between IME and a storage backend explicitly. Decisions about data locality will not be made for a whole (and potentially big) workflow. Instead, the system will make decisions by looking ahead to several steps of the workflow, allowing reacting to the observed dynamics of the execution. Ultimately, when a user-script runs, the information about the intended I/O schedule is communicated from EIOS through a modified filename, which is then used by the ESDM-aware application to determine the data placement.

4.4. Modified Workflow Execution

The steps to execute a workflow enriched with I/O information and perform smarter scheduling is depicted in Fig. 8. Components of EIOS are involved in different steps of the workflow and the I/O path. The suggested alterations can be seen in boxes pointed by red arrows, and the remaining components are the current state-of-the-art for workflows in climate and weather from Fig. 2. In the following, we describe the modifications we propose in this vision paper for each component involved in the software stack.

1. **Scientist** The user now has to provide an additional file that covers the I/O information for each task and slight changes have to be made to the current scripts.