Pantheon workflows with E4S

PI: David Rogers (LANL)

Project Coordinator: Terry Turton (LANL)

LA-UR-21-20358
Pantheon and E4S support end-to-end ECP examples

**Overview:** The Exascale Computing Project (ECP) is a complex undertaking, involving a myriad of technologies working together. An outstanding need is a way to capture, curate, communicate and validate workflows that cross all of these boundaries.

The Pantheon and E4S projects are collaborating to advance the integration and testing of capabilities, and to promote understanding of the complex workflows required by the ECP project. Utilizing a host of ECP technologies (spack, Ascent, Cinema, among others), this collaboration brings curated workflows to the fingertips of ECP researchers.

**Contributions**
- Curated end-to-end application/in-situ analysis examples can be run quickly by anyone on Summit. ([GitHub](https://github.com/pantheonscience/ECP-E4S-Examples))
- Pantheon/E4S integration speeds up build/setup times over source builds due to cached binaries (approx. 10x speed up).
E4S promotes progress on Reproducibility and Useability

• Reproducibility
  – isolating builds (dependencies, etc.)
  – targeting workflows to specific tasks and metrics

• Useability
  – Minimizing pull/build/run time, to make iteration useful and experimentation possible
  – Making workflows as self-documenting as possible
    • A knowledgeable user can dig through the code and understand what’s going on
    • Start with the `execute` script ...

E4S supports both reproducibility and useability for all types of complex workflows
What is a Pantheon workflow?
Pantheon Project vision:

A permanent working record of milestones. These workflows can serve both as records of milestone work, and examples for future work.

A set of reproducible workflow examples. Creating ECP-based workflows is complex, so working examples are a valuable product.

Reproducibility for exploratory research. Early software development efforts have unique build requirements and suffer from inconsistent versioning practices. Pantheon can capture the messy state of a working experiment.
Pantheon supports the following **states** for a workflow. Each is expected to be appropriately supported (through bug fixes and feature development), and have active project communication available to the community.

- **Release** The workflow is guaranteed to work in a production setting, as defined by the project. Supports bug fixes, feature development, and community outreach.
  - This link is used to designate a release workflow.
- **Development** The workflow is headed to the **release** state, but is not guaranteed to work on all target platforms. This type of workflow is generally provided to allow visibility and access into leading edge workflows, so that the community can use current and late-breaking features as a starting point for their own work.
  - This link is used to designate a development workflow.
- **Research** The workflow is an example that is of use to the community, but may not be supported at all times. A **research** workflow is not required to be headed towards a development/release state. It is not required to be continuously updated and tested on target platforms. Publishers are expected to actively field questions about the workflow, so that it can be of use to the broader community.
  - This link is used to designate a research workflow.
ECP-E4S-Examples Repository

https://github.com/pantheonscience/ECP-E4S-Examples

A set of working examples of end-to-end pipelines

- Ascent mini-app, Cinema DB, verification
  - E4S, Spack
- Nyx visualization, Cinema database, verification
  - E4S, Spack
- Nyx sampling w/CUDA, Cinema database, verification
  - Spack-only
  - E4S, Spack (issue reported)
- SW4 post-processing workflow (early example pipeline)
- Prototype container-based effort with E4S and Spack
Using this repository

First, clone the repository, then:

- Log onto Summit
- In a shell:

```
  git clone git@github.com:pantheonscience/ECP-E4S-Examples.git
  cd ECP-E4S-Examples
  git submodule update --init --recursive
```

- Then, `cd` into the directory for the example you’d like to run
- Edit the `bootstrap.env` file to include your summit allocation ID
- Execute the workflow by typing `./execute`

When the workflow is run, the following files will be run in this order:

- `setup/install-deps.sh`
  - `setup/install-app.sh` (called from the above script)
- `run/run.sh` (this submits the job)
- `run/wait_for_completion.sh`
- `postprocessing/postprocessing.sh`
- `validate/validate.sh`
What are workflows of interest?

Workflows that:

- cross product boundaries
- embody test cases
- embody tasks

Currently, our focus is on post-processing visualization with Cinema
Current results

Reproducible workflows running on Summit (Collaboration with E4S and Pantheon project)
Provide repository-managed end-to-end ECP workflows. Using E4S provides 10x speedup in builds.

- Ascent mini-app, Cinema DB, verification
  - E4S, Spack
- Nyx visualization, Cinema database, verification
  - E4S, Spack
- Nyx sampling w/CUDA, Cinema database, verification
  - Spack-only
  - E4S, Spack (issue reported)
- SW4 post-processing workflow (early example pipeline)
- Prototype container-based effort with E4S, Spack and containers

End-to-end issues reported as a result of reproducible workflows
- Rendering artifact in Ascent reported to Ascent team. Team was already aware of issue. Now fixed.
- Voiced support for Spack feature request (issue 13528)
- Issue reported to Spack and Ascent teams: runtime error with cached Ascent build (under review)

Other collaborations
- Cinema python module now integrated into ParaView release 5.9 (in pre-release now)
- Cinema utilized as visualization framework for the Foresight project
- Newest cinema viewer (jupyter notebook) being used by Ascent team
How Does Cinema Fit into In-situ Analysis?
What is Cinema?

- **Cinema** is part of an integrated workflow, providing a method of extracting, saving, analyzing or modifying and viewing complex data artifacts from large scale simulations.
  - If you’re having difficulty exploring the complex results from your simulation, Cinema can help.

- **The Cinema ‘Ecosystem’** is an integrated set of writers, viewers, and algorithms that allow scientists to export, analyze/modify and view Cinema databases.
  - This ecosystem is embodied in widely used tools (**ParaView**, **VisIt**, **Ascent**) and the database specification.
Time scrolling with a Cinema database
Comparative Analysis with Cinema:View
Pantheon examples
Ascent mini-app to Cinema DB; verification

Reproducible workflows runs on Summit
• E4S, Spack, Ascent, Cinema in Pantheon workflow

Steps
• Creates build location
• Clones specific commit of Spack
• Builds specific commit of Ascent, including coupled app/in-situ workflows
• Runs workflow to produce a Cinema database
• Verifies database

Using Spack and E4S drastically speeds up pull/build/run iteration time.
Nyx to Cinema DB; verification

Reproducible workflows runs on Summit
• E4S, Spack, Ascent, Cinema in Pantheon workflow

Steps
• Creates build location
• Clones specific commit of Spack
• Uses Spack to build Ascent, pulling E4S cache
• Builds Nyx against Ascent.
• Runs workflow to produce a Cinema database
• Verifies database

Using Spack and E4S drastically speeds up pull/build/run iteration time.
Nyx sampling w/CUDA Cinema DB; verification

Reproducible workflows runs on Summit
• E4S, Spack, Ascent, Cinema in Pantheon workflow

Steps
• Creates build location
• Clones specific commit of Spack
• Uses Spack to build Ascent
  • optionally pulls E4S cache
• Builds Nyx against Ascent.
• Runs an Ascent sampling algorithm workflow to produce a Cinema database
• Verifies database

This workflow produced a runtime error with the E4S cache that did not occur with clean build. Spack and Ascent teams are investigating.
Reproducible workflows runs on Summit
• Early workflow, mostly specific commits and home rolled scripts

Steps
• Creates build location
• Clones Alpine-DAV version of SW4
• Clones Ascent
• Builds coupled code
• Installs Cinema components
• Runs coupled workflow to produce Cinema database
• Runs Cinema algorithm on DB
• Installs Cinema:Explorer viewer

This early workflow is an example of a post-processing Cinema database workflow.
Conclusion

Contact:

dhr@lanl.gov

pantheon@lanl.gov
Pantheon and E4S support end-to-end ECP examples

Overview: The Exascale Computing Project (ECP) is a complex undertaking, involving a myriad of technologies working together. An outstanding need is a way to capture, curate, communicate and validate workflows that cross all of these boundaries.

The Pantheon and E4S projects are collaborating to advance the integration and testing of capabilities, and to promote understanding of the complex workflows required by the ECP project. Utilizing a host of ECP technologies (spack, Ascent, Cinema, among others), this collaboration brings curated workflows to the fingertips of ECP researchers.

Contributions
- Curated end-to-end application/in-situ analysis examples can be run quickly by anyone on Summit. ([https://github.com/pantheonscience/ECP-E4S-Examples](https://github.com/pantheonscience/ECP-E4S-Examples))
- Pantheon/E4S integration speeds up build/setup times over source builds due to cached binaries (approx.10x speed up).
Questions?