Pantheon workflows with E4S

Business Sensitive Information

PI: David Rogers (LANL)

Project Coordinator: Terry Turton (LANL)

LA-UR-21-20358

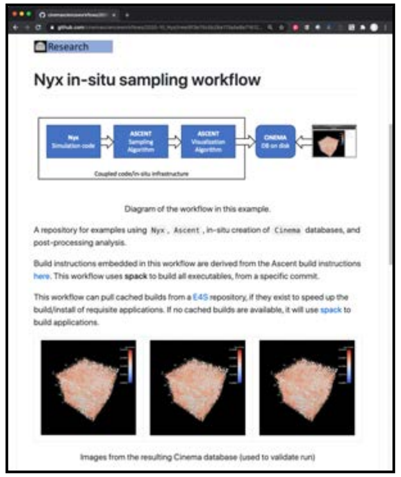
Pantheon and E4S support end-to-end ECP examples

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. (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).


Research

Nyx in-situ sampling workflow

Nyx Simulation code
ASCENT Sampling Algorithm
ASCENT Visualization Algorithm
CINEMA DB on disk
Coupled code/in-situ infrastructure

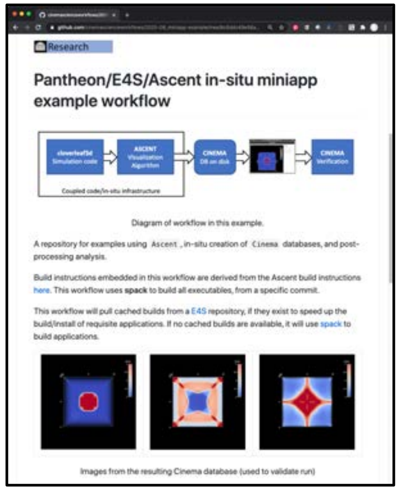
Diagram of the workflow in this example.

A repository for examples using `Nyx`, `Ascent`, in-situ creation of `Cinema` databases, and post-processing analysis.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow can pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

Images from the resulting Cinema database (used to validate run)


Research

Pantheon/E4S/Ascent in-situ miniapp example workflow

cloverleaf3d Simulation code
ASCENT Visualization Algorithm
CINEMA DB on disk
CINEMA Verification
Coupled code/in-situ infrastructure

Diagram of workflow in this example.

A repository for examples using `Ascent`, in-situ creation of `Cinema` databases, and post-processing analysis.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow will pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

Images from the resulting Cinema database (used to validate run)

Instructions page for (top) Nyx, Ascent and Cinema workflow repository, and (bottom) Cloverleaf3d, Ascent, Cinema workflow. These curated workflows use Pantheon, E4S and spack to provide curated workflows for ECP.

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

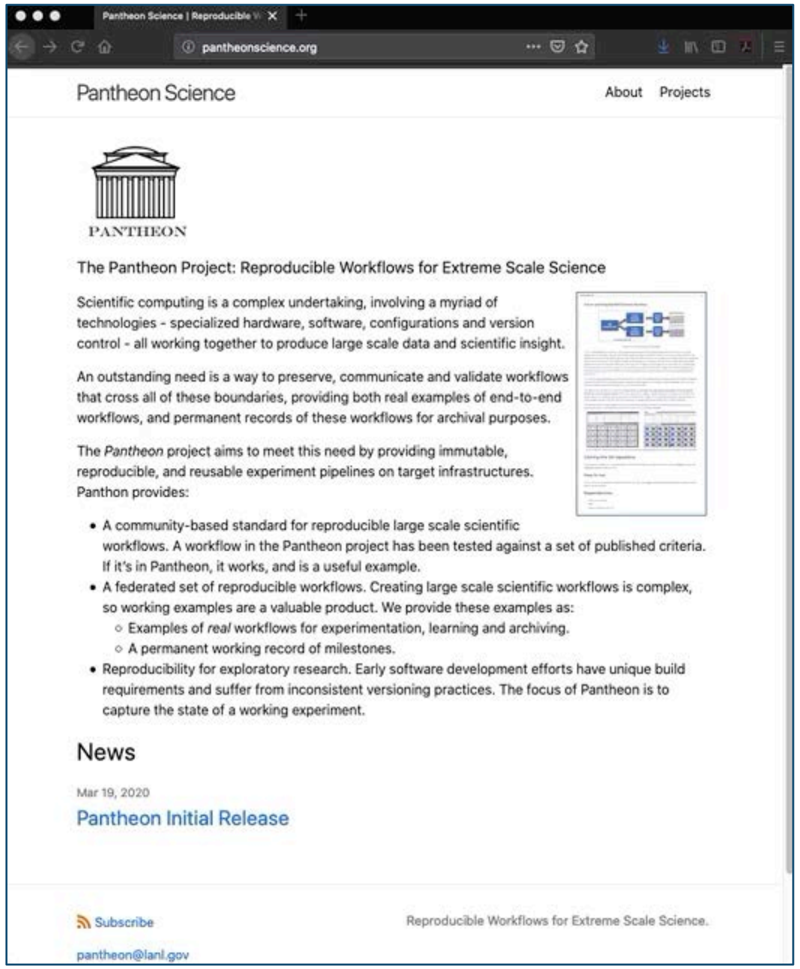
What is a Pantheon workflow?

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.



pantheonscience.org

Pantheon Science — About · Projects

The Pantheon Project: Reproducible Workflows for Extreme Scale Science

Scientific computing is a complex undertaking, involving a myriad of technologies - specialized hardware, software, configurations and version control - all working together to produce large scale data and scientific insight.

An outstanding need is a way to preserve, communicate and validate workflows that cross all of these boundaries, providing both real examples of end-to-end workflows, and permanent records of these workflows for archival purposes.

The *Pantheon* project aims to meet this need by providing immutable, reproducible, and reusable experiment pipelines on target infrastructures. Panthon provides:

- A community-based standard for reproducible large scale scientific workflows. A workflow in the Pantheon project has been tested against a set of published criteria. If it's in Pantheon, it works, and is a useful example.
- A federated set of reproducible workflows. Creating large scale scientific workflows is complex, so working examples are a valuable product. We provide these examples as:
  - Examples of *real* workflows for experimentation, learning and archiving.
  - A permanent working record of milestones.
- Reproducibility for exploratory research. Early software development efforts have unique build requirements and suffer from inconsistent versioning practices. The focus of Pantheon is to capture the state of a working experiment.

News

Mar 19, 2020

Pantheon Initial Release

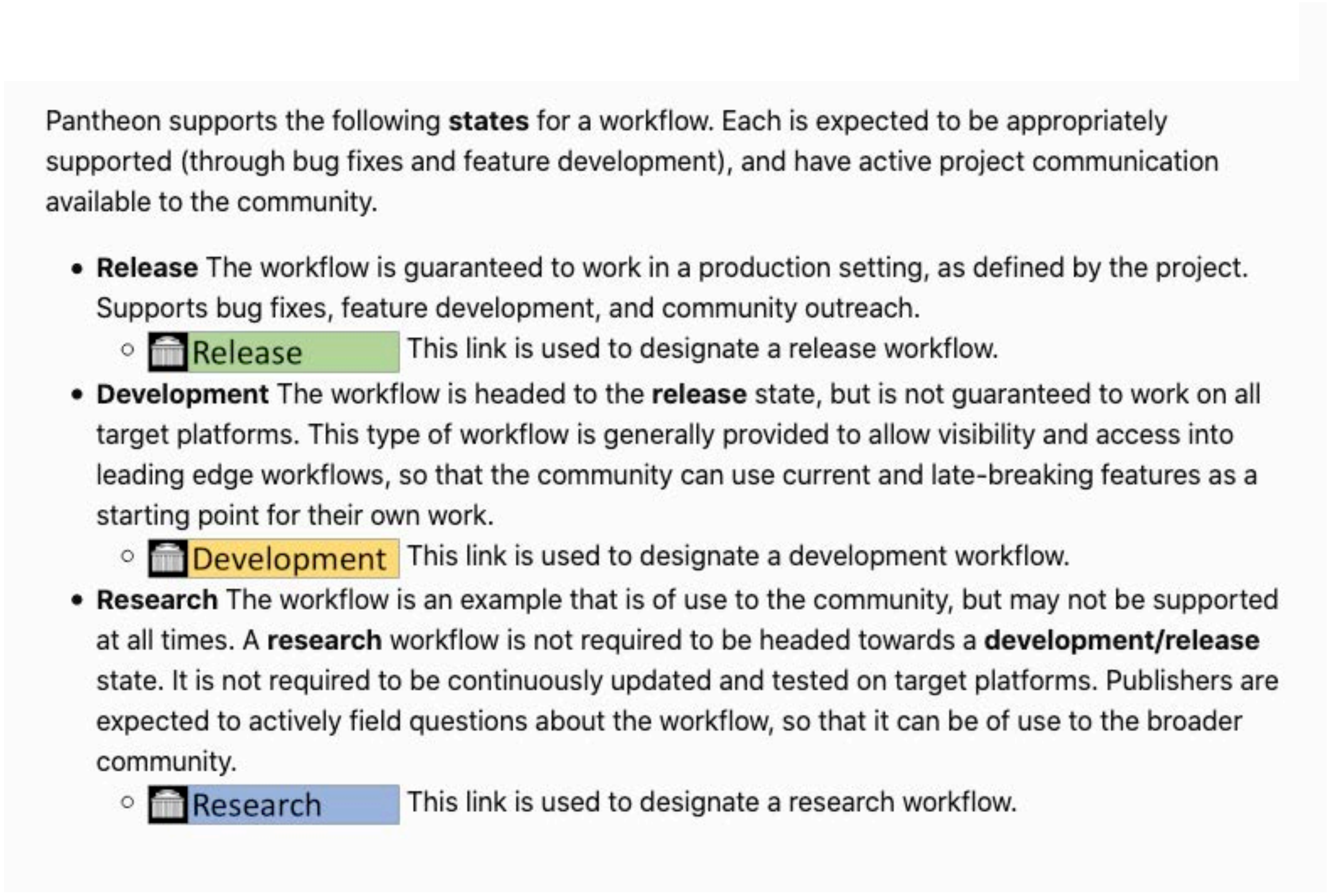
Subscribe

Reproducible Workflows for Extreme Scale Science.

pantheon@lanl.gov

Pantheon States

PANTHEON



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.

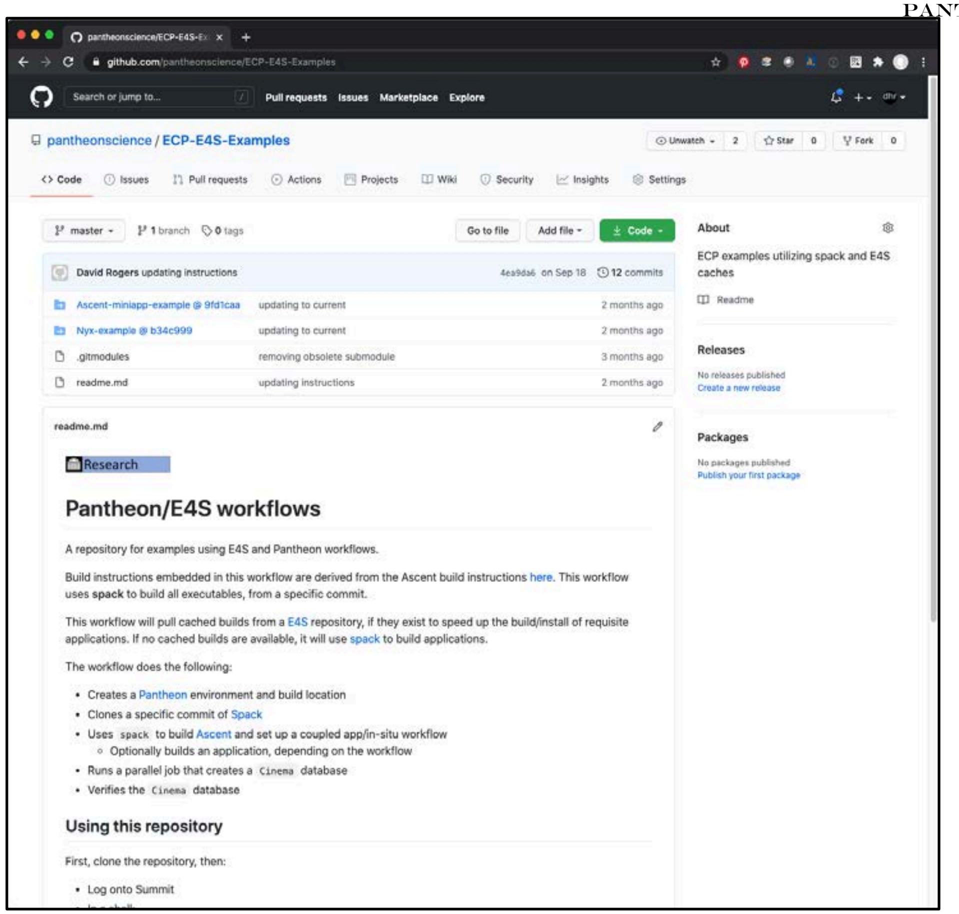
- **Release** The workflow is guaranteed to work in a production setting, as defined by the project. Supports bug fixes, feature development, and community outreach.
  - Release This link is used to designate a release workflow.
- **Development** The workflow is headed to the **release** state, but is 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.
  - Development 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.
  - Research 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



David Rogers updating instructions	4ea9da6 on Sep 18	12 commits
Ascent-miniapp-example @ 9fd1caa	updating to current	2 months ago
Nyx-example @ b34c999	updating to current	2 months ago
.gitmodules	removing obsolete submodule	3 months ago
readme.md	updating instructions	2 months ago

About

ECP examples utilizing spack and E4S caches

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

readme.md

Research

Pantheon/E4S workflows

A repository for examples using E4S and Pantheon workflows.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow will pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

The workflow does the following:

- Creates a Pantheon environment and build location
- Clones a specific commit of Spack
- Uses `spack` to build Ascent and set up a coupled app/in-situ workflow
    - Optionally builds an application, depending on the workflow
- Runs a parallel job that creates a `Cinema` database
- Verifies the `Cinema` database

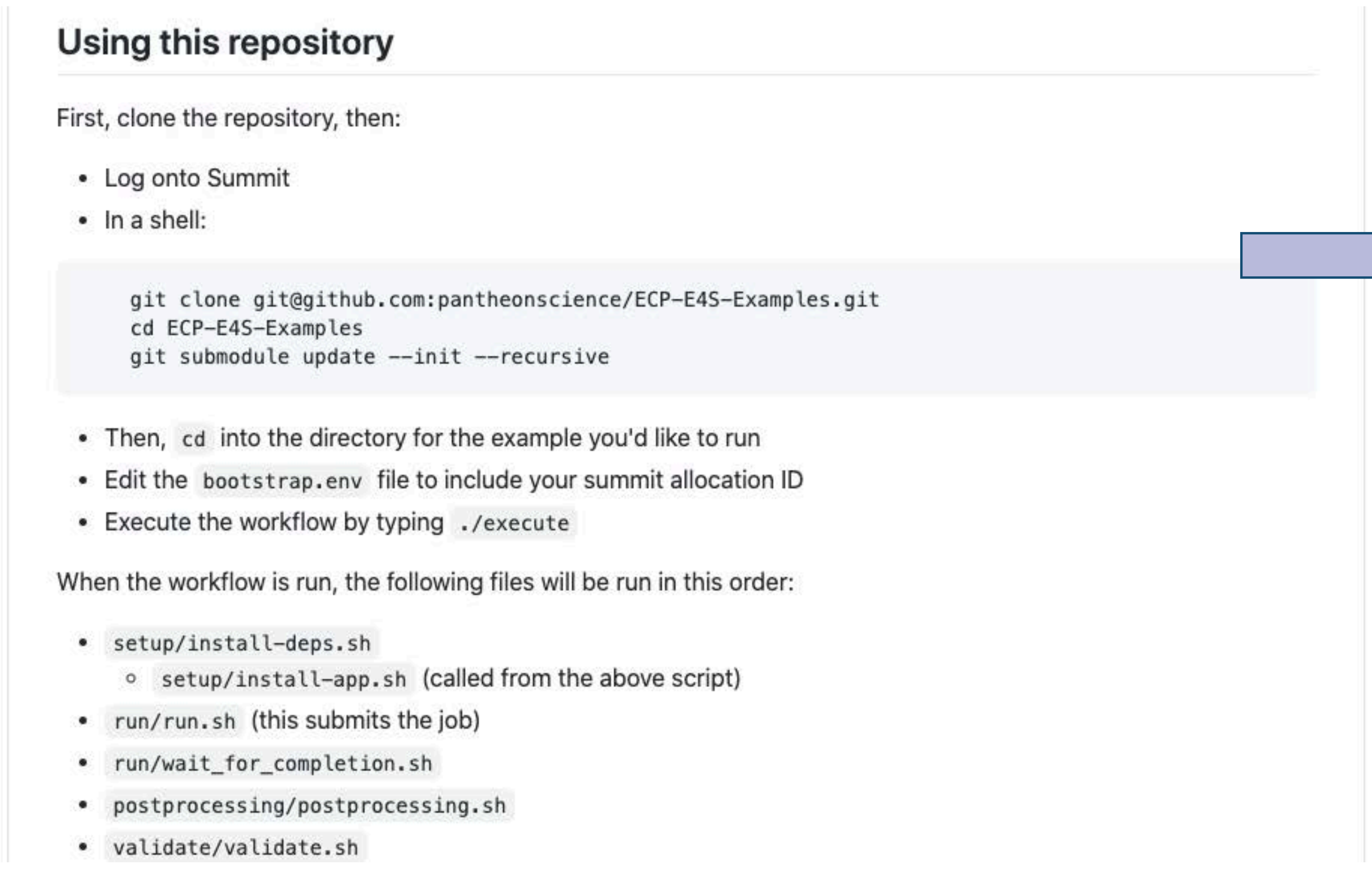
Using this repository

First, clone the repository, then:

- Log onto Summit

PANTHEON

ECP-E4S-Examples Repository (how to run)



Using this repository

First, clone the repository, then:

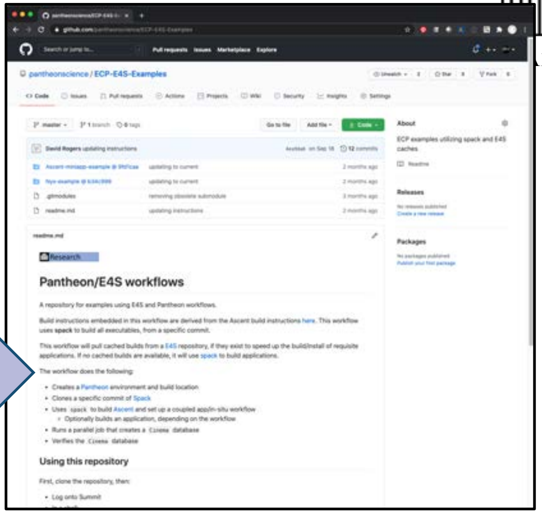
- Log onto Summit
- In a shell:

```
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`



pantheonscience / ECP-E4S-Examples

readme.md

Research

Pantheon/E4S workflows

A repository for examples using E4S and Pantheon workflows.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow will pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

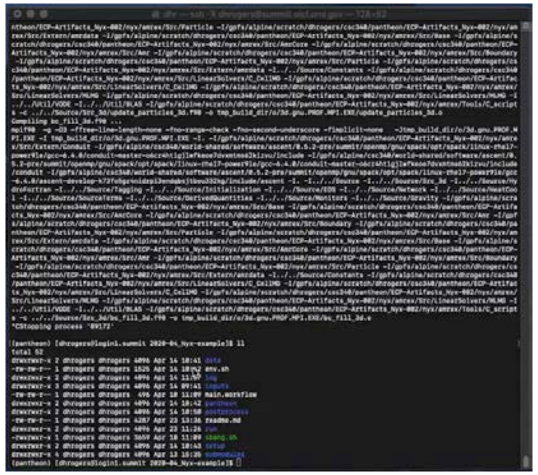
The workflow does the following:

- Creates a Pantheon environment and build location
- Clones a specific commit of Spack
- Uses `spack` to build Ascent and set up a coupled app/in-situ workflow
  - Optionally builds an application, depending on the workflow
- Runs a parallel job that creates a `Cinema` database
- Verifies the `Cinema` database

Using this repository

First, clone the repository, then:

- Log onto Summit



```
ntheon/ECP-Artifacts_Nyx-002/nyx/amrex/Src/Particle -I/gpfs/alpine/scratch/dhrogers/csc340/pantheon/ECP-Art
rex/Src/Extern/amrdata -I/gpfs/alpine/scratch/dhrogers/csc340/pantheon/ECP-Artifacts_Nyx-002/nyx/amrex/Src/
[illegible]
```

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

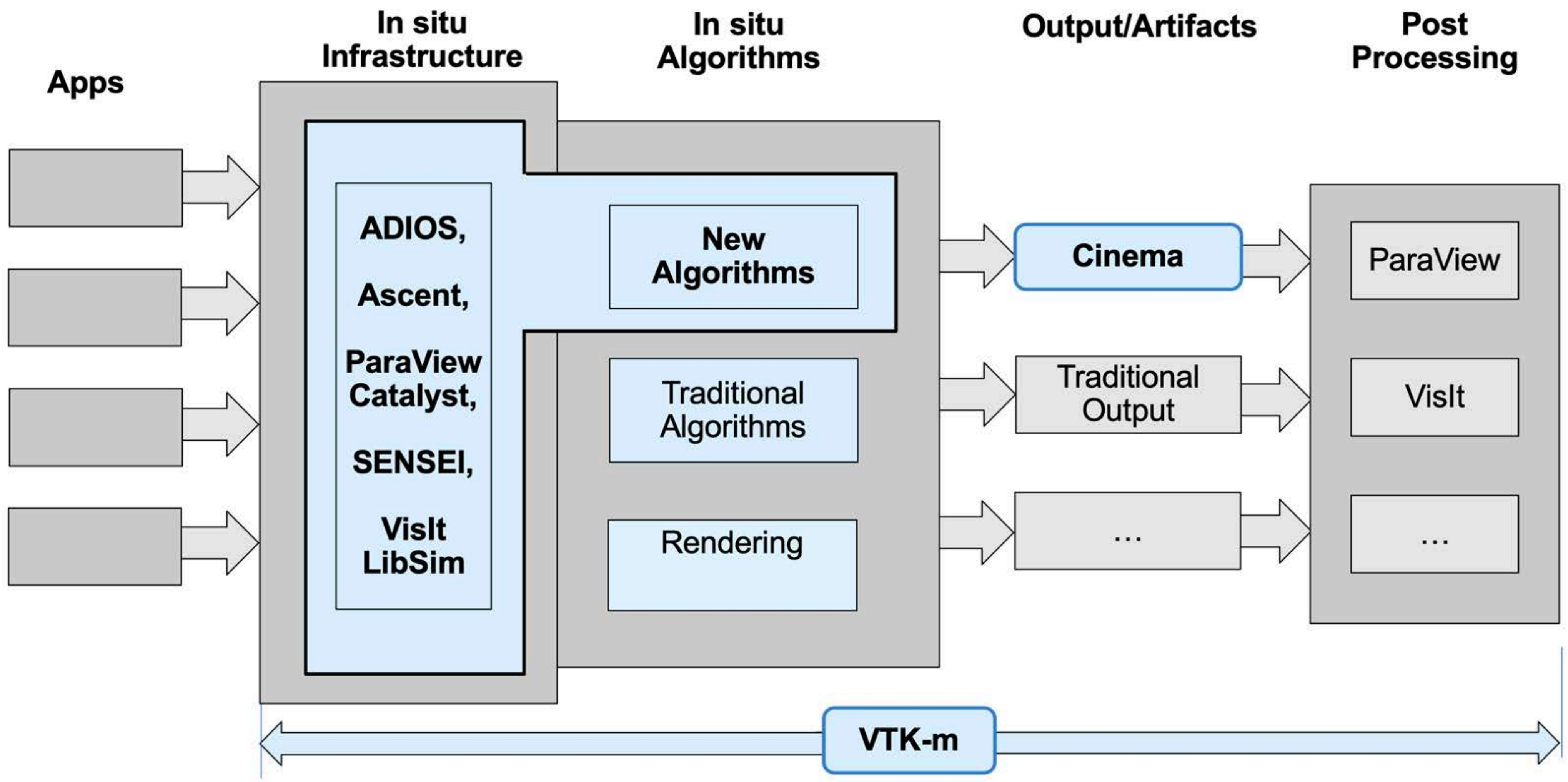
Reproducible workflows running on Summit (Collaboration with **E4S** and Pantheon project)
Provide repository-managed end-to-end ECP workflows. Using E4S provides 10x speedup in builds.

- 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)

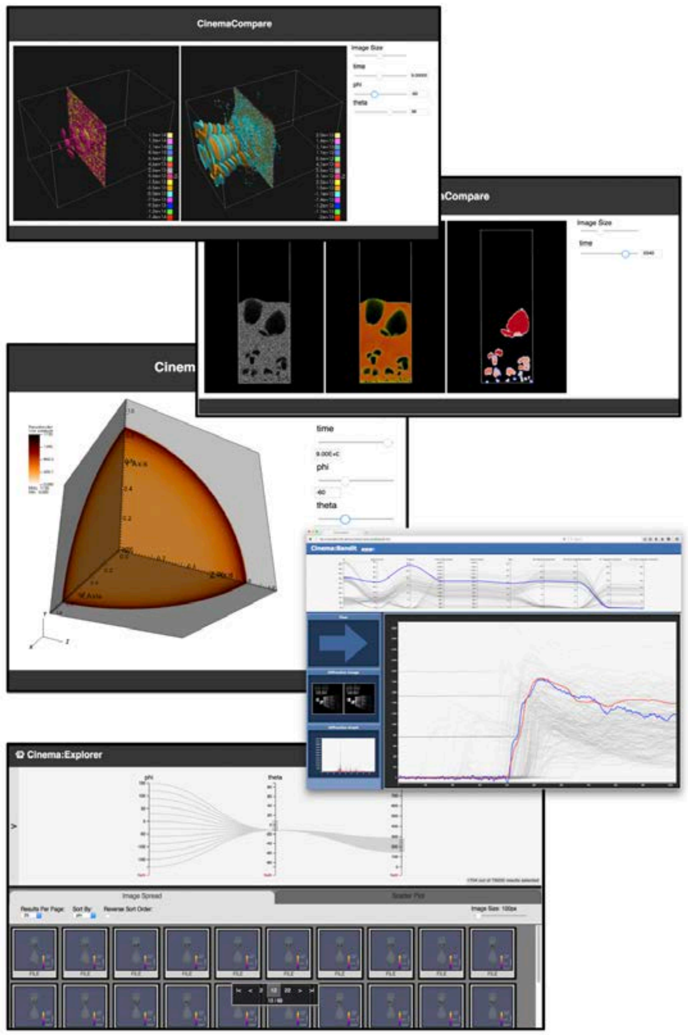
Other collaborations
- Cinema python module now integrated into ParaView release 5.9 (in pre-release now)
- Cinema utilized as visualization framework for the Foresight project
- Newest cinema viewer (jupyter notebook) being used by Ascent team

How Does Cinema Fit into In-situ Analysis?

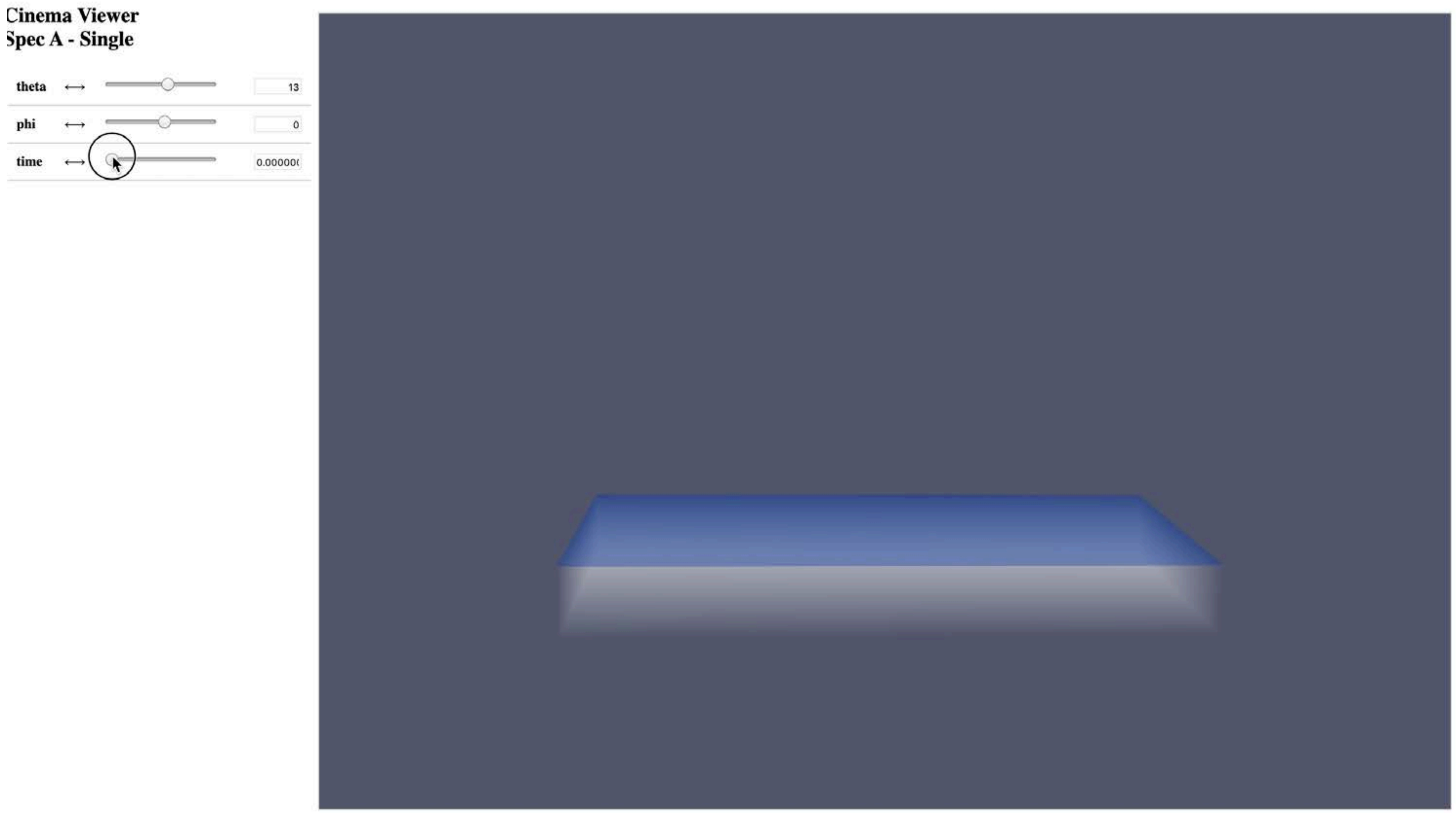


What is Cinema?

- **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.



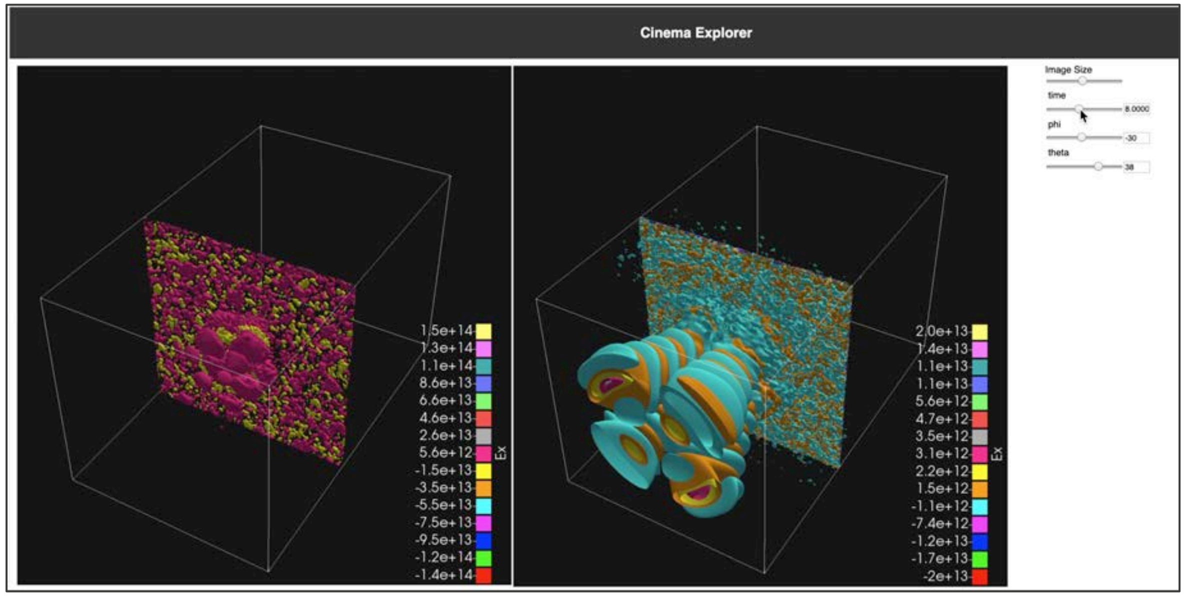
Time scrolling with a Cinema database



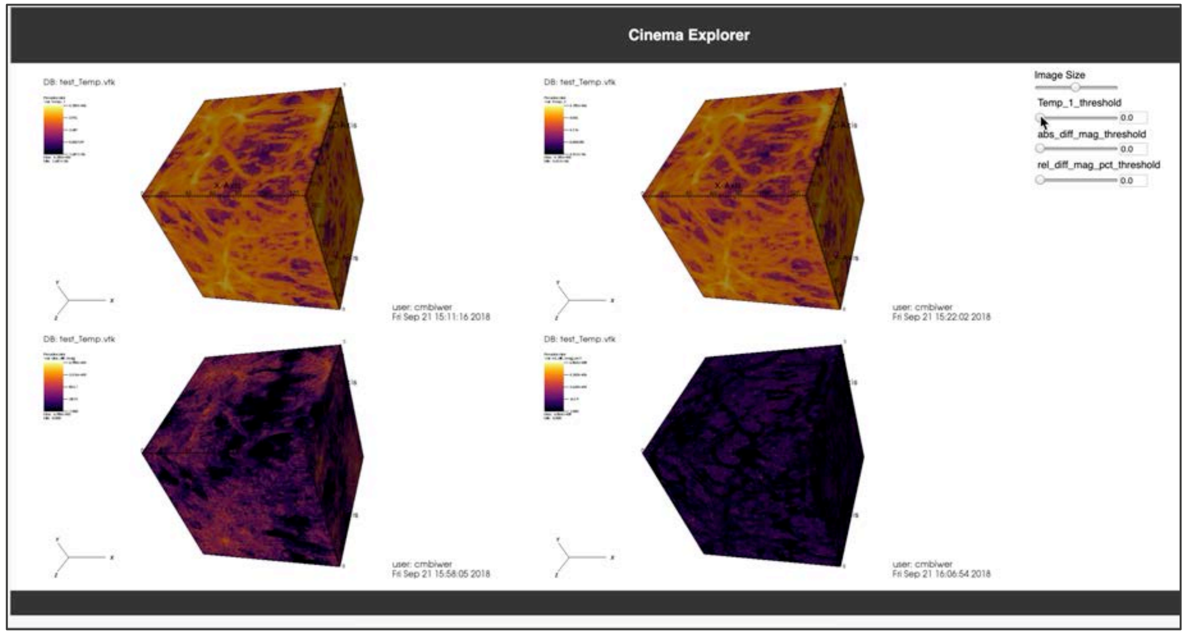
Cinema Viewer
Spec A - Single

theta	⟷	13
phi	⟷	0
time	⟷	0.000000

Comparative Analysis with Cinema:View



time 8.0000
phi -30
theta 38



user: cmbiwer
Fri Sep 21 15:11:16 2018

user: cmbiwer
Fri Sep 21 15:22:02 2018

user: cmbiwer
Fri Sep 21 15:58:05 2018

user: cmbiwer
Fri Sep 21 16:06:54 2018

Temp_1_threshold 0.0
abs_diff_mag_threshold 0.0
rel_diff_mag_pct_threshold 0.0

Pantheon examples

EXASCALE COMPUTING PROJECT

Ascent mini-app to Cinema DB; verification

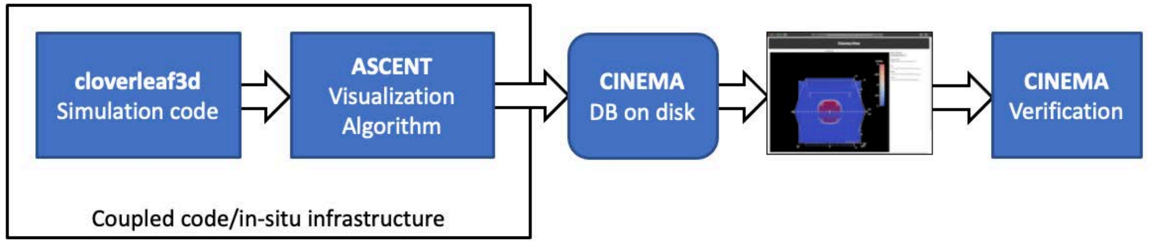
Reproducible workflows runs on Summit

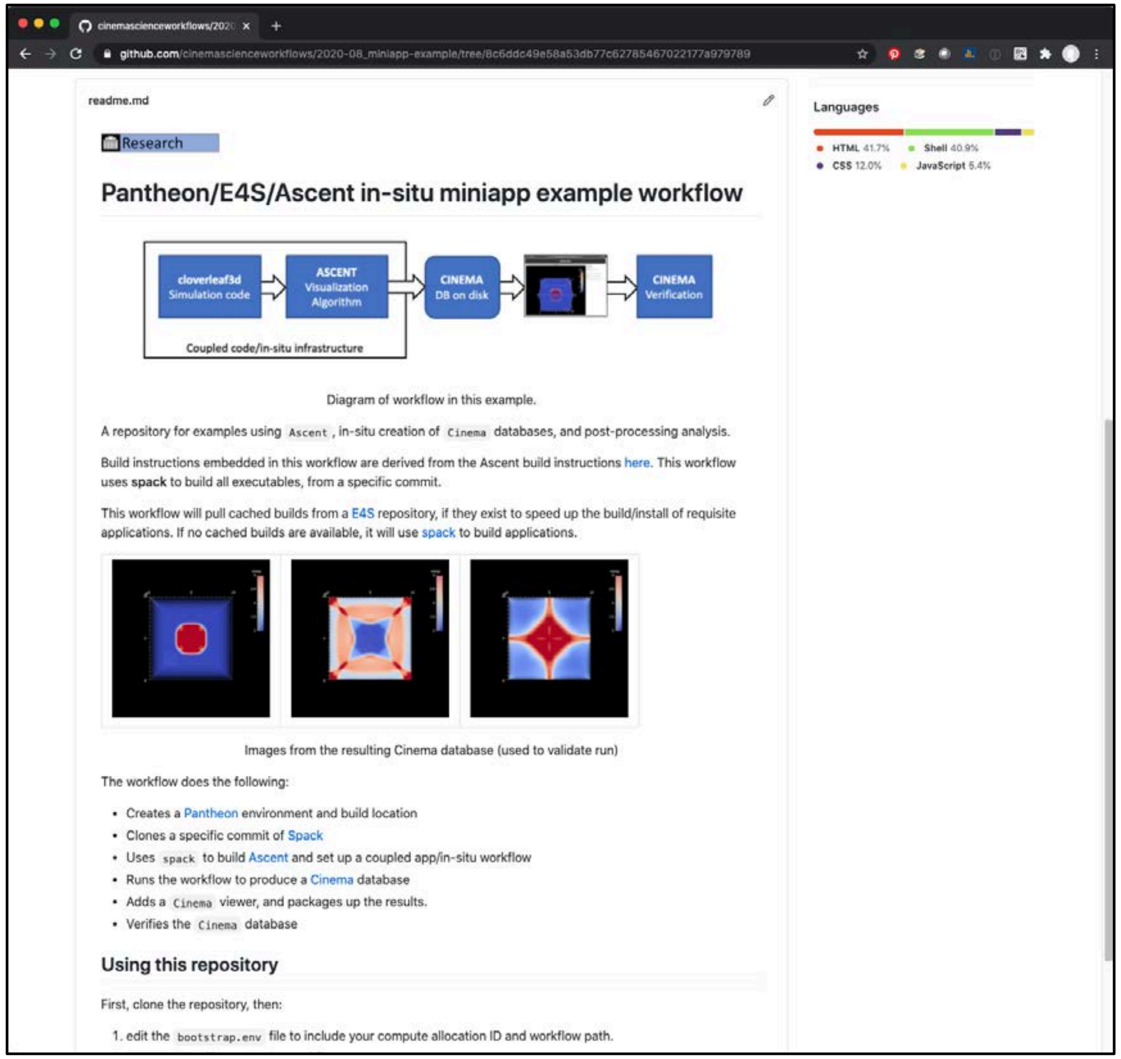
- E4S, Spack, Ascent, Cinema in Pantheon workflow

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.





cinemascienceworkflows/2020

github.com/cinemascienceworkflows/2020-08_miniapp-example/tree/8c6ddc49e58a53db77c627854670221 77a979789

readme.md

Research

Pantheon/E4S/Ascent in-situ miniapp example workflow

Diagram of workflow in this example.

A repository for examples using `Ascent`, in-situ creation of `Cinema` databases, and post-processing analysis.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow will pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

Images from the resulting Cinema database (used to validate run)

The workflow does the following:

- Creates a Pantheon environment and build location
- Clones a specific commit of Spack
- Uses `spack` to build Ascent and set up a coupled app/in-situ workflow
- Runs the workflow to produce a Cinema database
- Adds a `Cinema` viewer, and packages up the results.
- Verifies the `Cinema` database

Using this repository

First, clone the repository, then:

1. edit the `bootstrap.env` file to include your compute allocation ID and workflow path.

Languages

● HTML 41.7%	● Shell 40.9%
● CSS 12.0%	● JavaScript 5.4%

Web page from the Pantheon project website. This points to a specific commit in a Cinema workflow repository. Details about Pantheon project available at pantheonsciece.org

Nyx to Cinema DB; verification

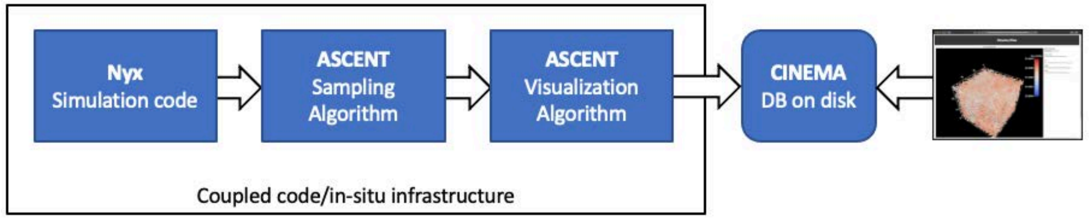
Reproducible workflows runs on Summit

- E4S, Spack, Ascent, Cinema in Pantheon workflow

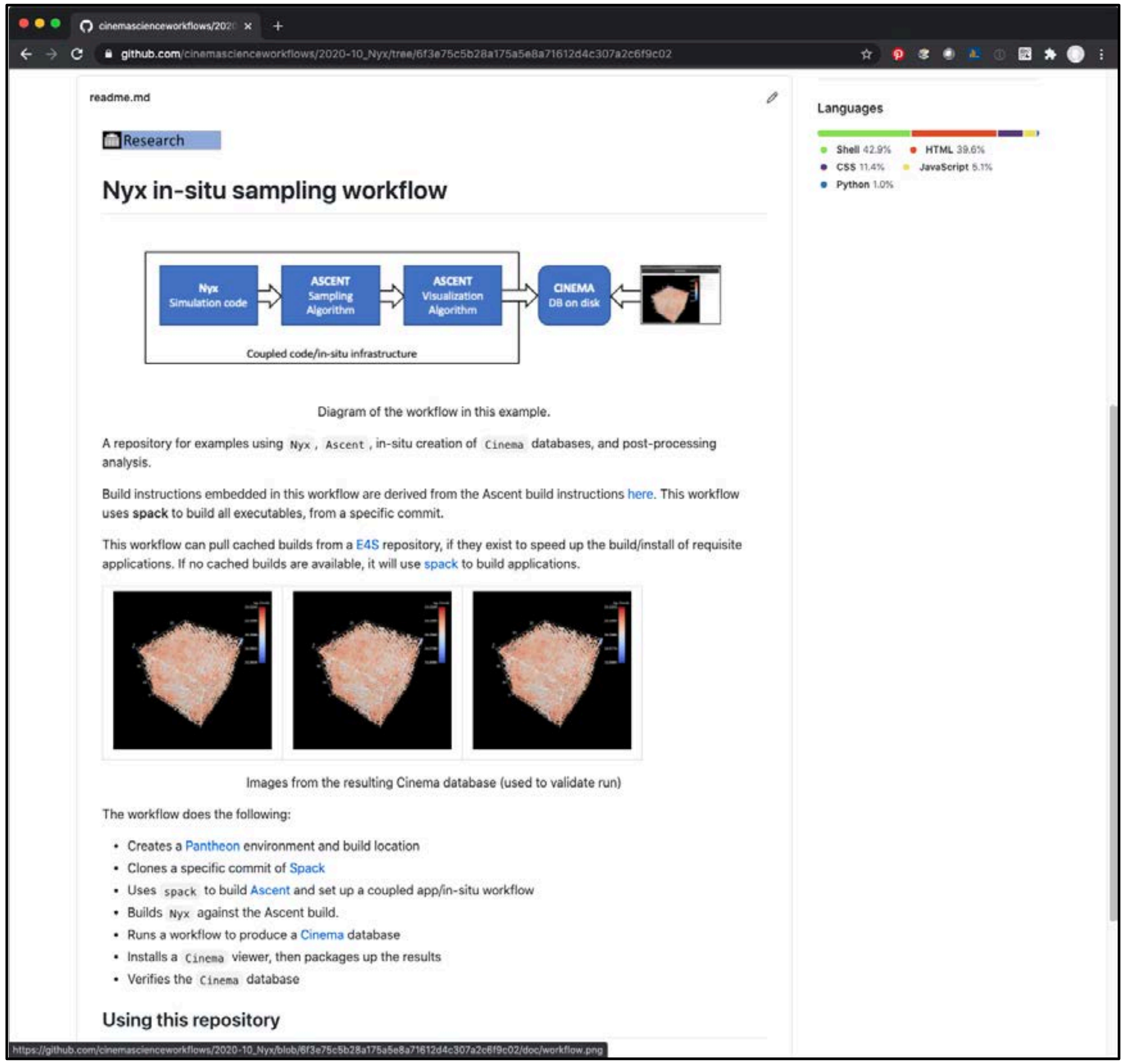
Steps

- 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 Simulation code
ASCENT Sampling Algorithm
ASCENT Visualization Algorithm
CINEMA DB on disk
Coupled code/in-situ infrastructure



cinemascienceworkflows/202
github.com/cinemascienceworkflows/2020-10_Nyx/tree/6f3e75c5b28a175a5e8a71612d4c307a2c6f9c02
readme.md
Research
Languages
Shell 42.9%
HTML 39.6%
CSS 11.4%
JavaScript 5.1%
Python 1.0%

Nyx in-situ sampling workflow

Diagram of the workflow in this example.

A repository for examples using `Nyx` , `Ascent` , in-situ creation of `Cinema` databases, and post-processing analysis.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow can pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

Images from the resulting Cinema database (used to validate run)

The workflow does the following:

- Creates a Pantheon environment and build location
- Clones a specific commit of Spack
- Uses `spack` to build Ascent and set up a coupled app/in-situ workflow
- Builds `Nyx` against the Ascent build.
- Runs a workflow to produce a Cinema database
- Installs a `Cinema` viewer, then packages up the results
- Verifies the `Cinema` database

Using this repository

https://github.com/cinemascienceworkflows/2020-10_Nyx/blob/6f3e75c5b28a175a5e8a71612d4c307a2c6f9c02/doc/workflow.png

Web page from the Pantheon project website. This points to a specific commit in a Cinema workflow repository. Details about Pantheon project available at pantheonsciece.org

EXASCALE COMPUTING PROJECT

Nyx sampling w/CUDA Cinema DB; verification

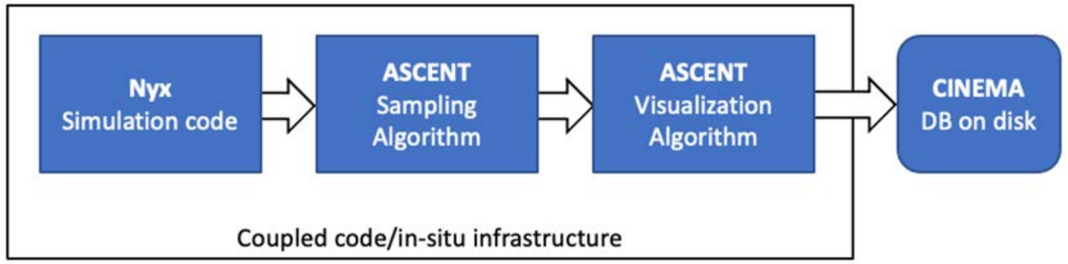
Reproducible workflows runs on Summit

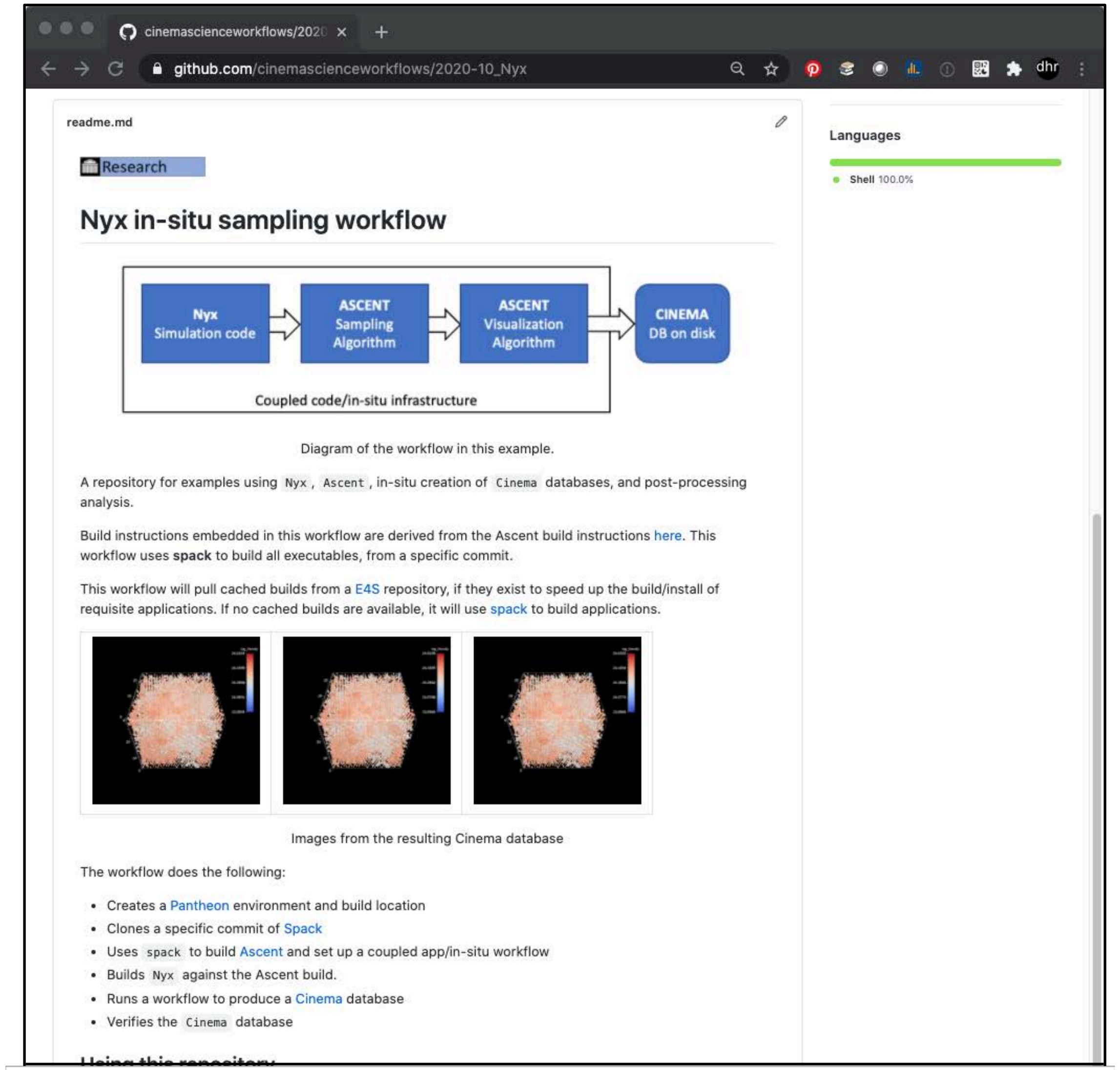
- E4S, Spack, Ascent, Cinema in Pantheon workflow

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.


Nyx Simulation code
ASCENT Sampling Algorithm
ASCENT Visualization Algorithm
CINEMA DB on disk
Coupled code/in-situ infrastructure


cinemascienceworkflows/2020
github.com/cinemascienceworkflows/2020-10_Nyx
readme.md
Research
Languages
Shell 100.0%
Nyx Simulation code
ASCENT Sampling Algorithm
ASCENT Visualization Algorithm
CINEMA DB on disk
Coupled code/in-situ infrastructure

Nyx in-situ sampling workflow

Diagram of the workflow in this example.

A repository for examples using `Nyx`, `Ascent`, in-situ creation of `Cinema` databases, and post-processing analysis.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow will pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

Images from the resulting Cinema database

The workflow does the following:

- Creates a Pantheon environment and build location
- Clones a specific commit of Spack
- Uses `spack` to build Ascent and set up a coupled app/in-situ workflow
- Builds `Nyx` against the Ascent build.
- Runs a workflow to produce a Cinema database
- Verifies the `Cinema` database

SW4 to Cinema DB; Cinema analysis; verification

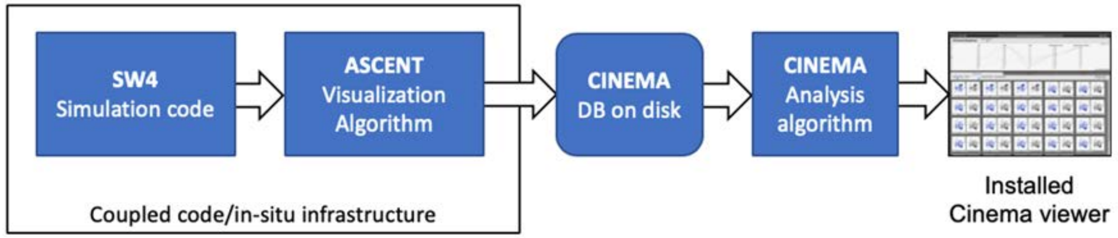
Reproducible workflows runs on Summit

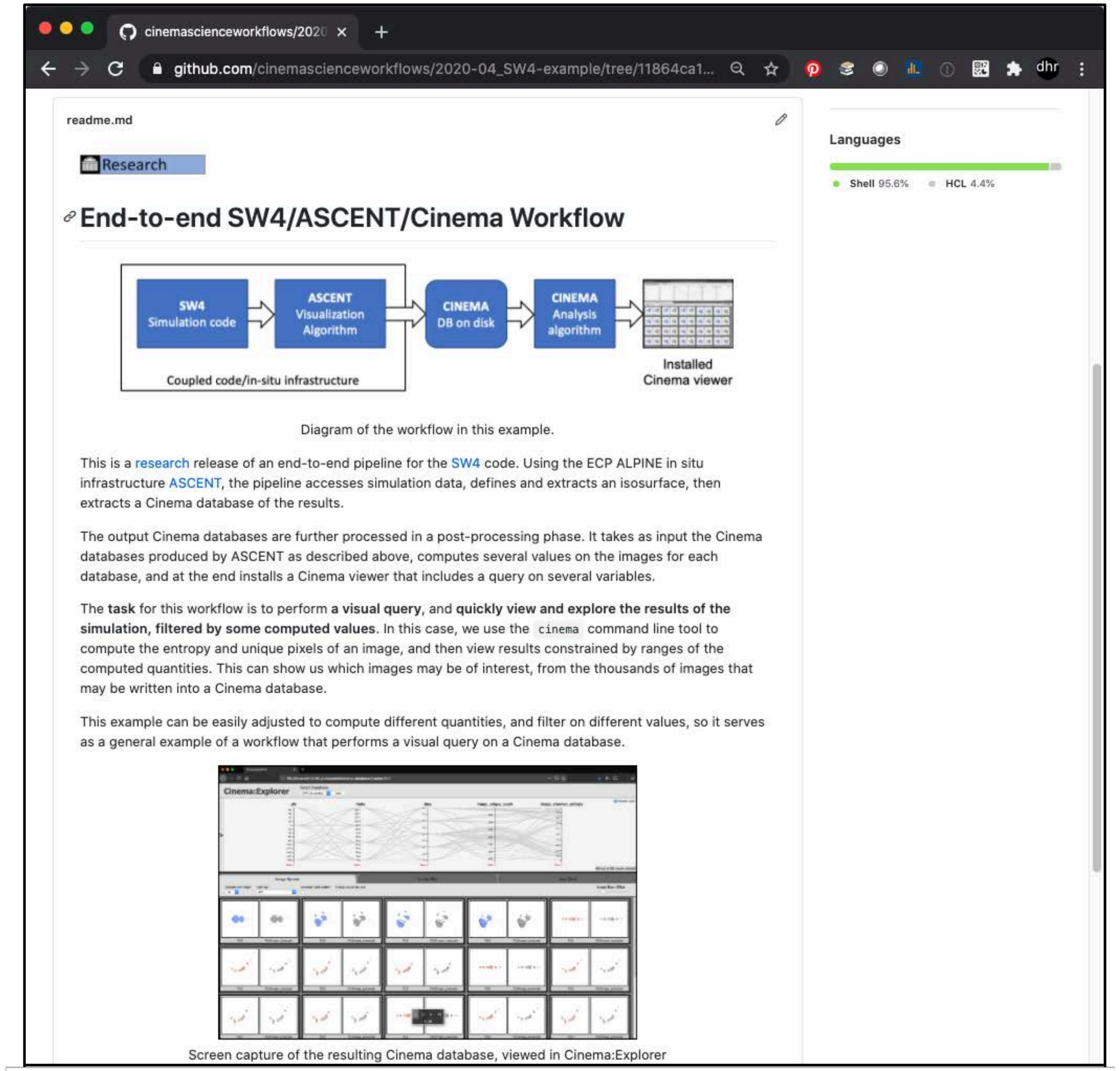
- Early workflow, mostly specific commits and home rolled scripts

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.




cinemascienceworkflows/2020 ×

github.com/cinemascienceworkflows/2020-04_SW4-example/tree/11864ca1...

readme.md

Research

Languages

Shell 95.6% HCL 4.4%

End-to-end SW4/ASCENT/Cinema Workflow

Diagram of the workflow in this example.

This is a research release of an end-to-end pipeline for the SW4 code. Using the ECP ALPINE in situ infrastructure ASCENT, the pipeline accesses simulation data, defines and extracts an isosurface, then extracts a Cinema database of the results.

The output Cinema databases are further processed in a post-processing phase. It takes as input the Cinema databases produced by ASCENT as described above, computes several values on the images for each database, and at the end installs a Cinema viewer that includes a query on several variables.

The **task** for this workflow is to perform **a visual query**, and **quickly view and explore the results of the simulation, filtered by some computed values**. In this case, we use the `cinema` command line tool to compute the entropy and unique pixels of an image, and then view results constrained by ranges of the computed quantities. This can show us which images may be of interest, from the thousands of images that may be written into a Cinema database.

This example can be easily adjusted to compute different quantities, and filter on different values, so it serves as a general example of a workflow that performs a visual query on a Cinema database.

Screen capture of the resulting Cinema database, viewed in Cinema:Explorer

Conclusion

Contact:

dhr@lanl.gov

pantheon@lanl.gov

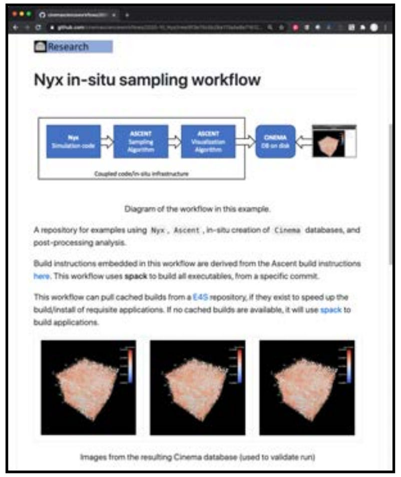
Pantheon and E4S support end-to-end ECP examples

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)
- Pantheon/E4S integration speeds up build/setup times over source builds due to cached binaries (approx.10x speed up).


Research

Nyx in-situ sampling workflow

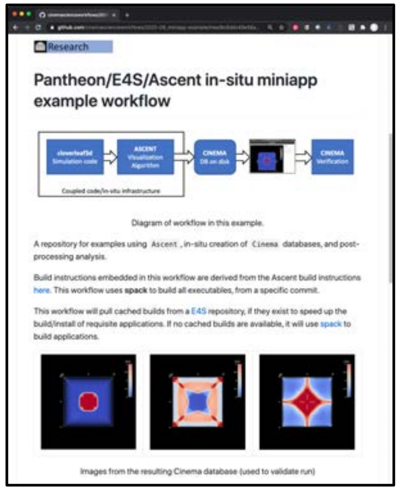
Diagram of the workflow in this example.

A repository for examples using `Nyx`, `Ascent`, in-situ creation of `Cinema` databases, and post-processing analysis.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow can pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

Images from the resulting Cinema database (used to validate run)


Research

Pantheon/E4S/Ascent in-situ miniapp example workflow

Diagram of workflow in this example.

A repository for examples using `Ascent`, in-situ creation of `Cinema` databases, and post-processing analysis.

Build instructions embedded in this workflow are derived from the Ascent build instructions here. This workflow uses **spack** to build all executables, from a specific commit.

This workflow will pull cached builds from a E4S repository, if they exist to speed up the build/install of requisite applications. If no cached builds are available, it will use spack to build applications.

Images from the resulting Cinema database (used to validate run)

Instructions page for (top) Nyx, Ascent and Cinema workflow repository, and (bottom) Cloverleaf3d, Ascent, Cinema workflow. These curated workflows use Pantheon, E4S and spack to provide curated workflows for ECP.

EXASCALE COMPUTING PROJECT

PANTHEON

E4S

Ascent

CINEMA

ECP COMPUTING PROJECT

Questions?