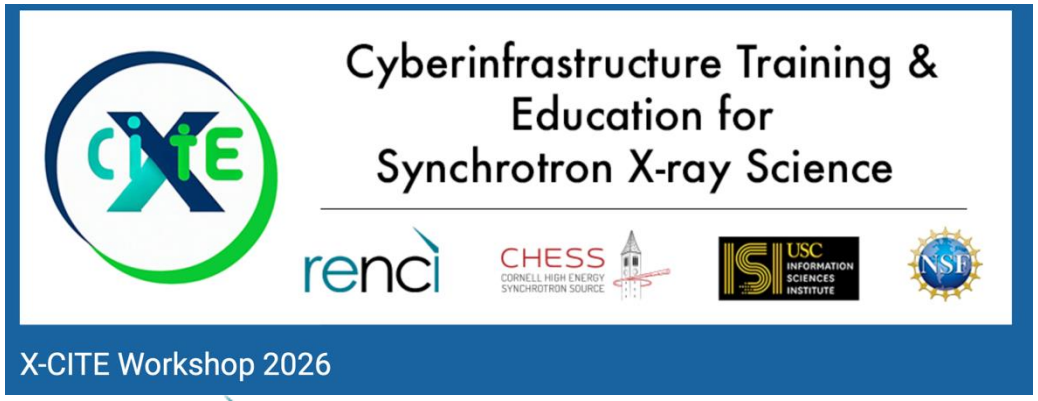
Funded by the National Science Foundation, Grant #2320373

Scientific Gateways, Open OnDemand, Workflows

Karan Vahi

USC Information Sciences Institute



NSF OAC Award Numbers: 2320373, 2320375, 2320374

Funded by the National Science Foundation, Grant #2320373

What Are Science Gateways?

- Community-developed platforms that give scientists, engineers, and students browser-based access to computing, data, and other research resources

- Provide a web-based interface that hides the complexity of the underlying cyberinfrastructure (CI) — researchers focus on the science, not on running HPC jobs or managing distributed data

- Gateways take many forms:
  - Web portals — submit and monitor jobs on HPC clusters
  - Domain-specific platforms (genomics, climate, materials science)
  - Workflow execution environments and data management platforms

- Examples: **Open OnDemand, Galaxy, ACCESS Pegasus, nanoHUB, CIPRES**

Funded by the National Science Foundation, Grant #2320373

How Science Gateways Help Researchers

- **Ease of access**
  - Replace SSH clients, command-line use, scheduler syntax (SLURM, SGE, PBS), and manual scp transfers with intuitive browser interfaces — no software to install
- **Domain-specific interfaces**
  - Interfaces tailored to specific scientific workflows (e.g., protein structure analysis), hiding the complexity of running them on HPC clusters
- **Democratization of computing**
  - Opens advanced resources to researchers at smaller institutions, students and early-career researchers, domain scientists without a computational background, and occasional users of national resources
- **Reproducibility and collaboration**
  - Shared workflows, provenance tracking (what ran, when, and with what inputs), and community-contributed tools and pipelines

Funded by the National Science Foundation, Grant #2320373

Open OnDemand: Browser-Based HPC Access

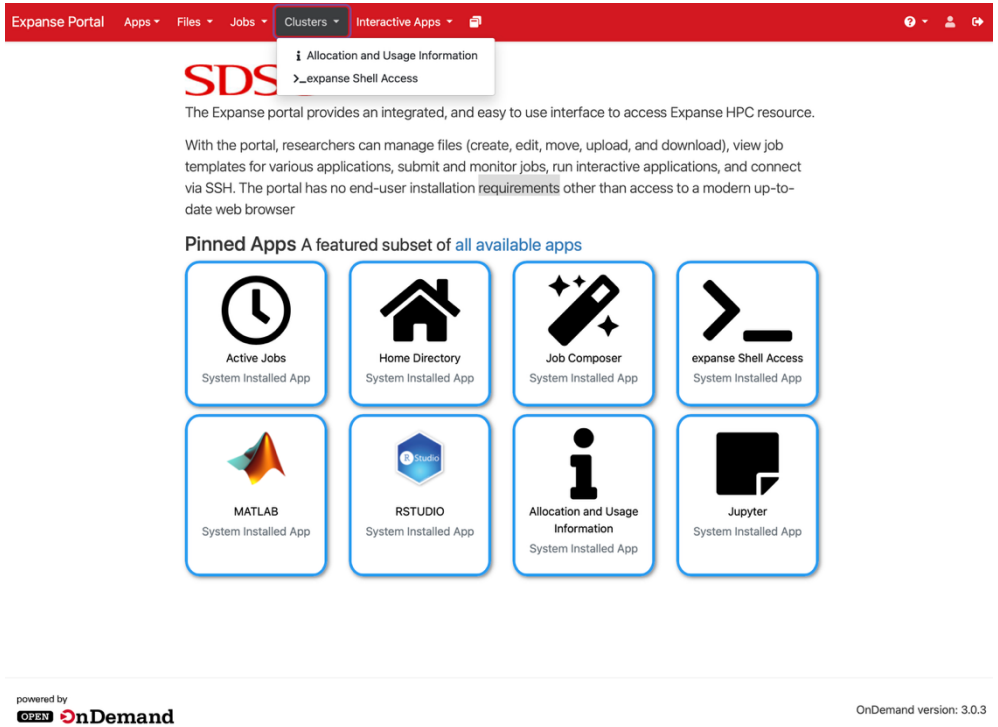
- Open-source, browser-based platform from the Ohio Supercomputer Center (OSC) — the standard web portal on most ACCESS resources

- Access HPC clusters with no client software: browse files, submit and monitor batch jobs, and launch interactive apps, all in your browser

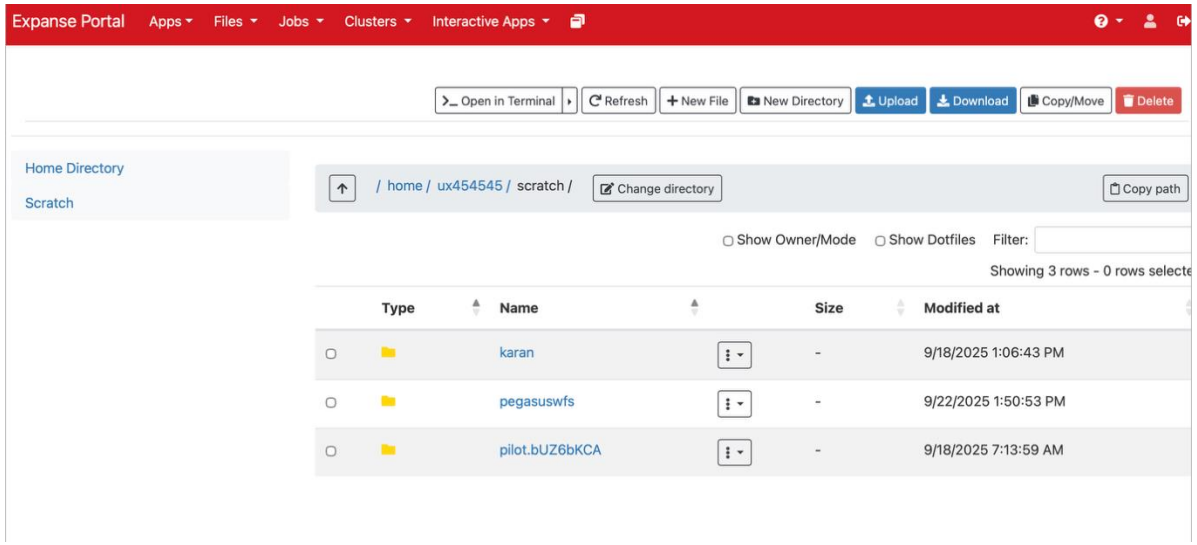
- **Key features**
  - **File Manager** — browse the filesystem and upload, download, or edit files in the browser (no SFTP or scp)
  - **Job Composer** — form-based job creation with reusable templates; no scheduler syntax to memorize
  - **Interactive Apps** — launch Jupyter, RStudio, MATLAB, or a remote desktop on compute nodes
  - **Shell Access** — web terminal to the login node, no SSH client (handy with 2-factor login)

Funded by the National Science Foundation, Grant #2320373

Open OnDemand on SDSC Expanse



c1



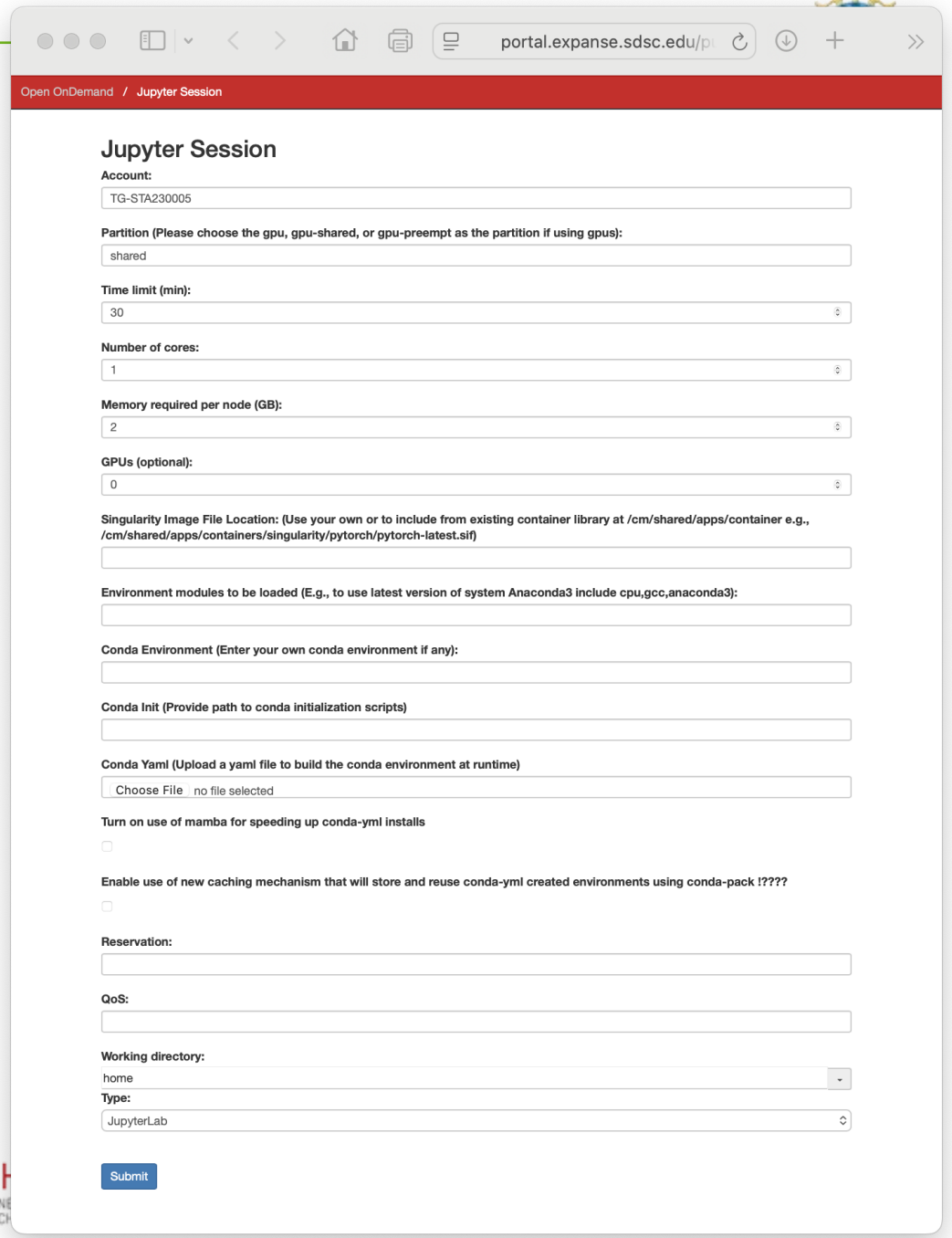
c2

Launching a Jupyter Notebook via Open OnDemand – SDSC Expanse

1. Go to the resource's portal URL and log in with your ACCESS credentials
2. Click Interactive Apps, then select Jupyter Notebook
3. Fill in the job form — account, partition (shared or compute), cores, memory, and walltime
4. Click Launch; wait for the job to move from Queued to Running
5. Click Connect to Jupyter to open the notebook in a new tab

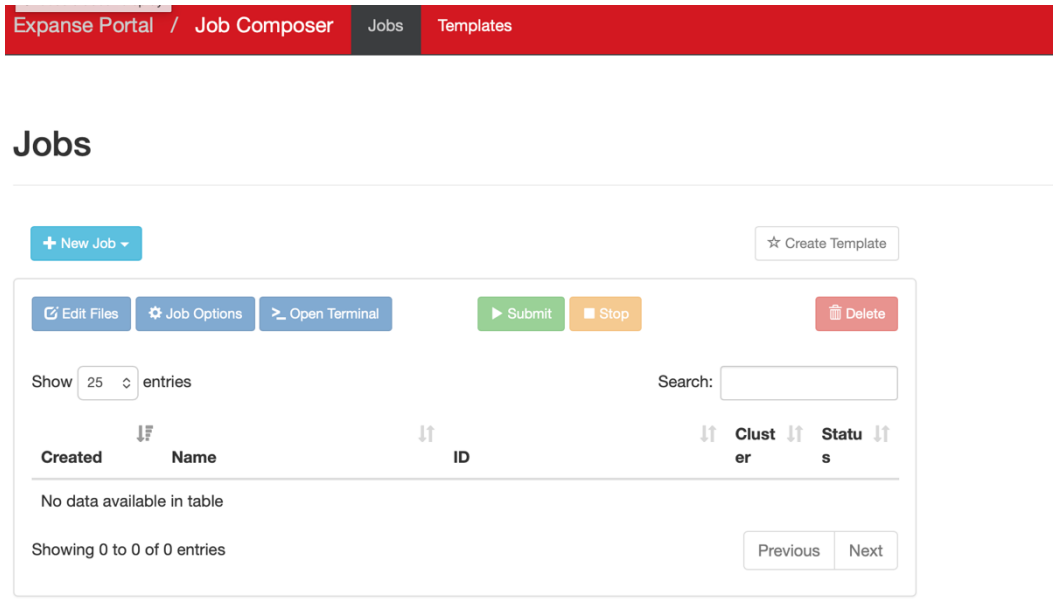
Tip: interactive jobs consume allocation credits — request only what you need and close your session when done.

ACCESS Resource	Open OnDemand URL
Anvil (Purdue)	ondemand.anvil.rcac.purdue.edu
Bridges-2 (PSC)	ondemand.bridges2.psc.edu
Delta (NCSA)	login.delta.ncsa.illinois.edu
Expanse (SDSC)	portal.expanse.sdsc.edu



Funded by the National Science Foundation, Grant #2320373

Open OnDemand: Job Composer



Build and submit jobs without the command line

The Job Composer builds jobs from reusable templates, then submits and monitors them — New Job, Edit Files, Submit / Stop, and a live status table, all in the browser.

Source: X-CITE — "Using Science Gateways" (xcitecourse.org)

Funded by the National Science Foundation, Grant #2320373

Open OnDemand: Anvil Demo

Login: https://ondemand.anvil.rcac.purdue.edu

Funded by the National Science Foundation, Grant #2320373

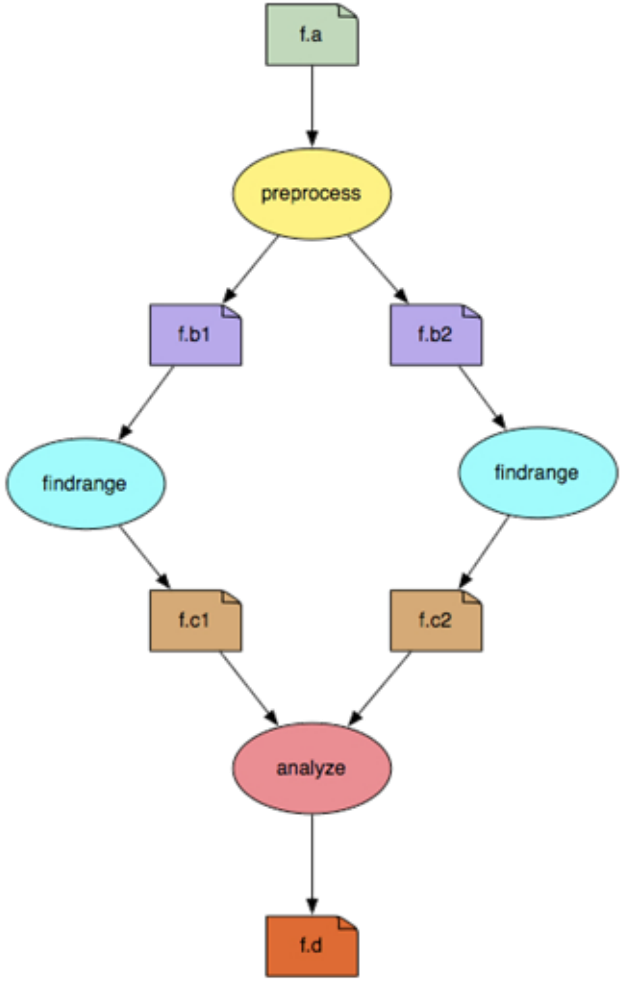
ACCESS Pegasus: A Gateway for Workflows

- A ready-to-use environment for running **Pegasus Workflows** on ACCESS resources — no need to install or configure Pegasus yourself
- Built on Open OnDemand and maintained by the Pegasus team at USC's Information Sciences Institute (USC/ISI)
- What it provides:
  - A pre-configured Pegasus submit host connected to ACCESS
  - Direct connectivity to Expanse, Anvil, Bridges-2, and other ACCESS resources, plus the OSG OSPool for high-throughput workloads
  - Notebooks run on pegasus.access-ci.org while jobs execute on HTCondor points — a clean separation of notebook and compute environments
- **Why use it:** zero configuration, multi-site execution, fault tolerance (retry and checkpointing), full provenance, and expert support from the Pegasus team

Funded by the National Science Foundation, Grant #2320373

Scientific Workflows

- An abstraction to express ensemble of complex computational operations
  - Eg: retrieving data from remote storage services, executing applications, and transferring data products to designated storage sites
- A workflow is represented as a directed acyclic graph (DAG)
  - Nodes: tasks or jobs to be executed
  - Edges: depend between the tasks
- Have a monolithic application/experiment?
- The tasks in a scientific workflow can be everything from short serial tasks to very large parallel tasks (MPI for example) surrounded by a large number of small, serial tasks used for pre- and post-processing.
- Find the inherent DAG structure in your application to convert into a workflow



Funded by the National Science Foundation, Grant #2320373

Workflow Challenges Across Domains

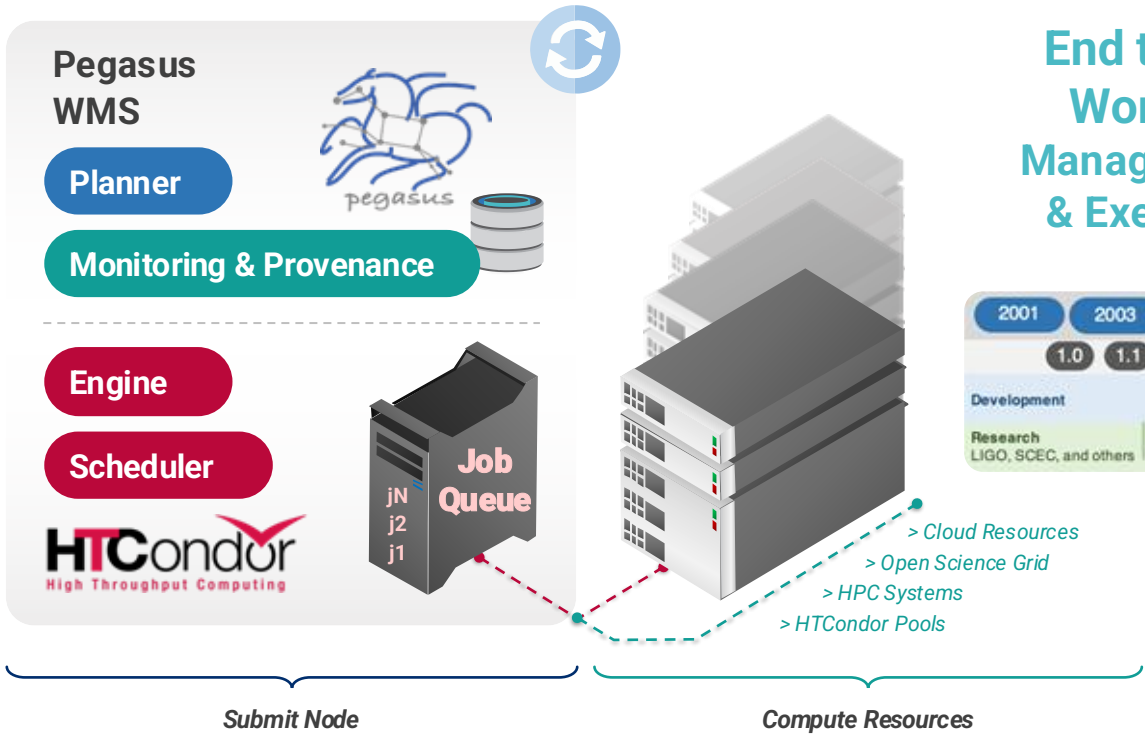
- Describe complex workflows in a simple way
- Access distributed, heterogeneous data and resources (heterogeneous interfaces)
- Deal with resources/software that change over time
- Ease of use. Ability to debug and monitor large workflows

Our Focus

- Separation between workflow description and workflow execution
- Workflow planning and scheduling (scalability, performance)
- Task execution (monitoring, fault tolerance, debugging, web dashboard)
- Provide additional assurances that a scientific workflow is not accidentally or maliciously tampered with during its execution.

renci USC Viterbi School of Engineering Information Sciences Institute CHESS CORNELL HIGH ENERGY SYNCHROTRON SOURCE

Pegasus Workflow Management System



End to End Workflow Management & Execution

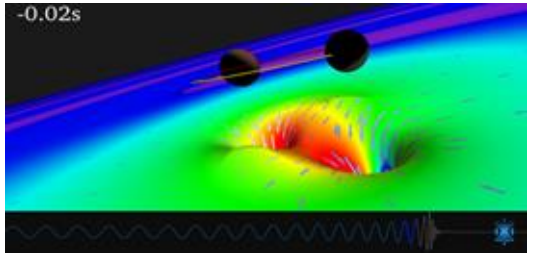
- Develop portable scientific workflows in Python, Java, and R
- Compile workflows to be run on heterogeneous resources
- Monitor and debug workflow execution via CLI and web-based tools
- Recover from failures with built-in fault tolerance mechanisms
- Regular release schedule incorporating latest research and development

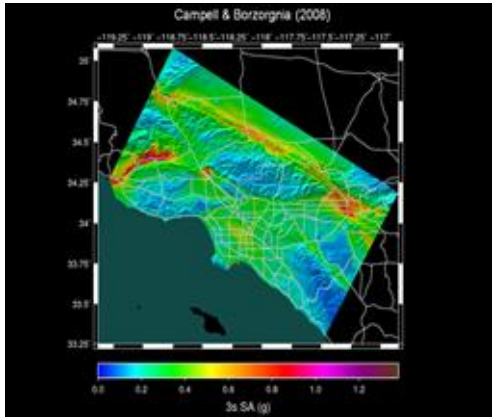
Pegasus in practice

- Laser Interferometer Gravitational Wave Observatory (LIGO) develops large scale analysis pipelines used for gravitational wave detection.
- Southern California Earthquake Center (SCEC) CyberShake project generates hazard maps using hierarchical workflows .
- The XENONnT project uses Pegasus for processing and monte carlo workflows, searching for dark matter

The XENONnT detector

LIGO observation of colliding black holes

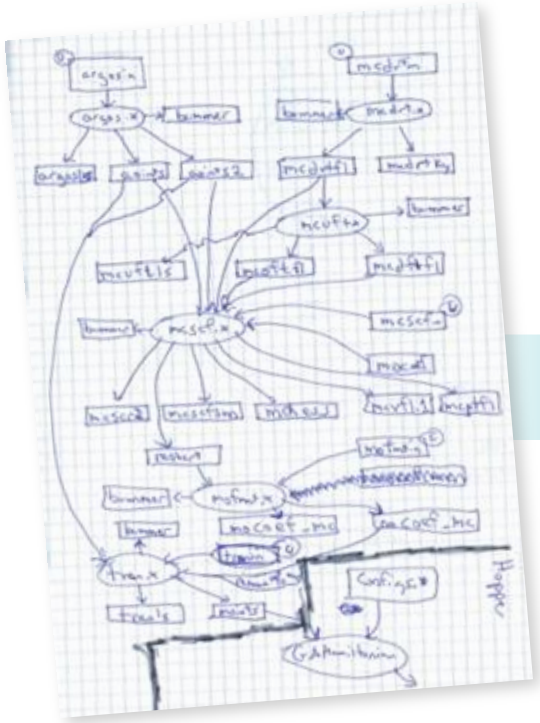




Hazard map indicating maximum amount of shaking at a particular geographic location generated from SCEC's CyberShake Pegasus workflow

Funded by the National Science Foundation, Grant #2320373

Pegasus provides APIs to generate the Abstract Workflow



```python
#!/usr/bin/env python3

import os
import logging
from pathlib import Path
from argparse import ArgumentParser

logging.basicConfig(level=logging.DEBUG)

# --- Import Pegasus API ------------------
from Pegasus.api import *

# --- Create Abstract Workflow ------------
wf = Workflow("pipeline")

webpage = File("pegasus.html")

# --- Create Parent Job -------------
curl_job = (
    Job("curl")
    .add_args("-o", webpage, "http://pegasus.isi.edu")
    .add_outputs(webpage, stage_out=False, register_replica=False)
)

count = File("count.txt")

# --- Create Dependent Job ----------
wc_job = (
    Job("wc")
    .add_args("-l", webpage)
    .add_inputs(webpage)
    .set_stdout(count, stage_out=True, register_replica=True)
)
# --- Add jobs to the Abstract Workflow --------
wf.add_jobs(curl_job, wc_job)

# --- Add control flow dependency -----------
wf.add_dependency(wc_job, parents=[curl_job])

# --- Write out the Abstract Workflow -----------
wf.write()
```

python

Java

jupyter

Abstract Workflow

```yaml
x-pegasus:
  apiLang: python
  createdBy: vahi
  createdOn: 11-19-20T14:57:58Z
pegasus: '5.0'
name: pipeline
jobs:
- type: job
  name: curl
  id: ID0000001
  arguments:
  - -o
  - pegasus.html
  - http://pegasus.isi.edu
  uses:
  - lfn: pegasus.html
    type: output
    stageOut: false
    registerReplica: false
- type: job
  name: wc
  id: ID0000002
  stdout: count.txt
  arguments:
  - -l
  - pegasus.html
  uses:
  - lfn: count.txt
    type: output
    stageOut: true
    registerReplica: true
  - lfn: pegasus.html
    type: input
jobDependencies:
- id: ID0000001
  children:
  - ID0000002
```

YAML Formatted

renci

USC Viterbi School of Engineering Information Sciences Institute

CHESS CORNELL HIGH ENERGY SYNCHROTRON SOURCE

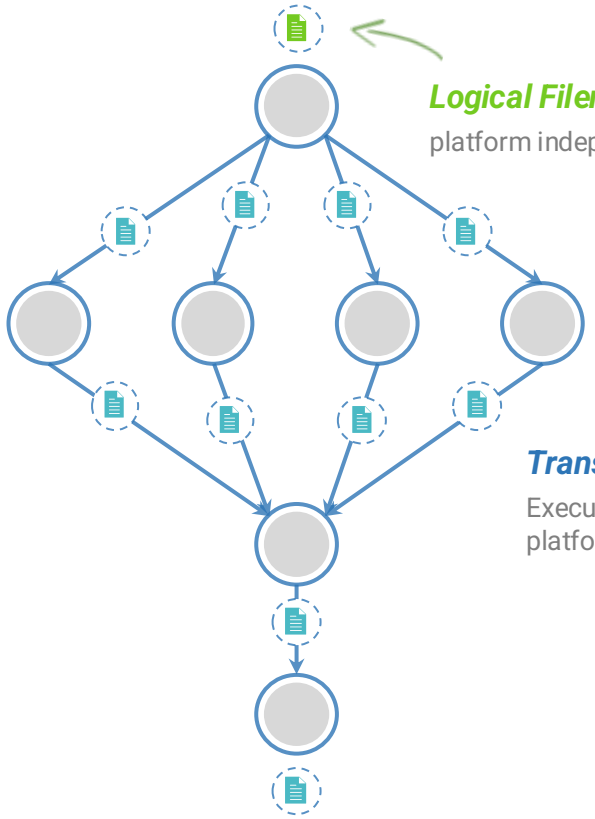
Funded by the National Science Foundation, Grant #2320373

Input Workflow Specification YAML formatted

Portable Description

Users do not worry about low level execution details

ABSTRACT WORKFLOW



Logical Filename (LFN)

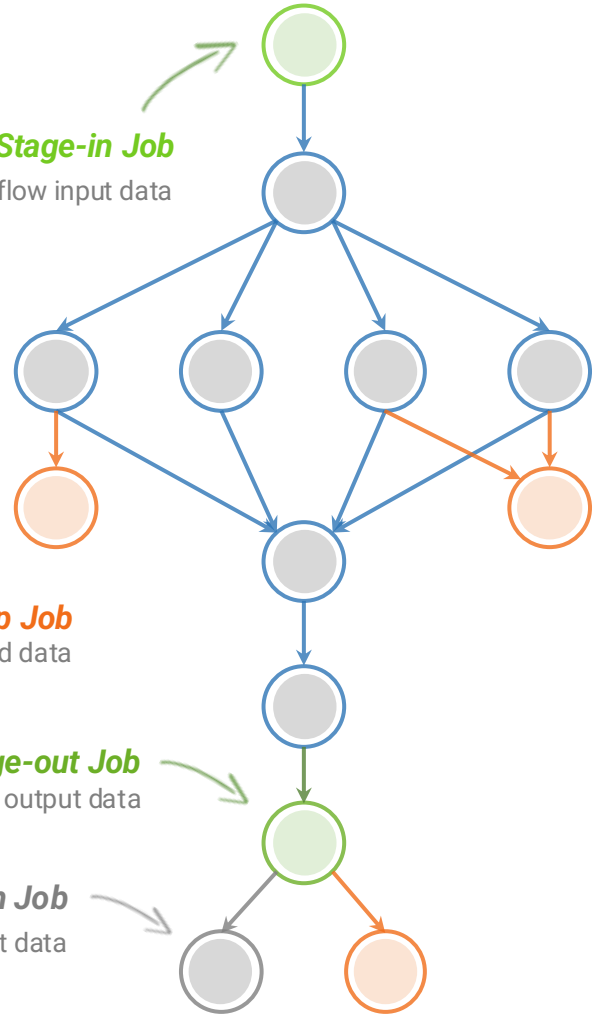
platform independent (abstraction)

Transformation

Executables (or programs)
platform independent

Output Workflow

EXECUTABLE WORKFLOW



Stage-in Job

Transfers the workflow input data

Cleanup Job

Removes unused data

Stage-out Job

Stage-out generated output data

Registration Job

Registers the workflow output data

renci

USC Viterbi School of Engineering Information Sciences Institute

CHESS CORNELL HIGH ENERGY SYNCHROTRON SOURCE

Funded by the National Science Foundation, Grant #2320373

Pegasus CHESS QM Workflow

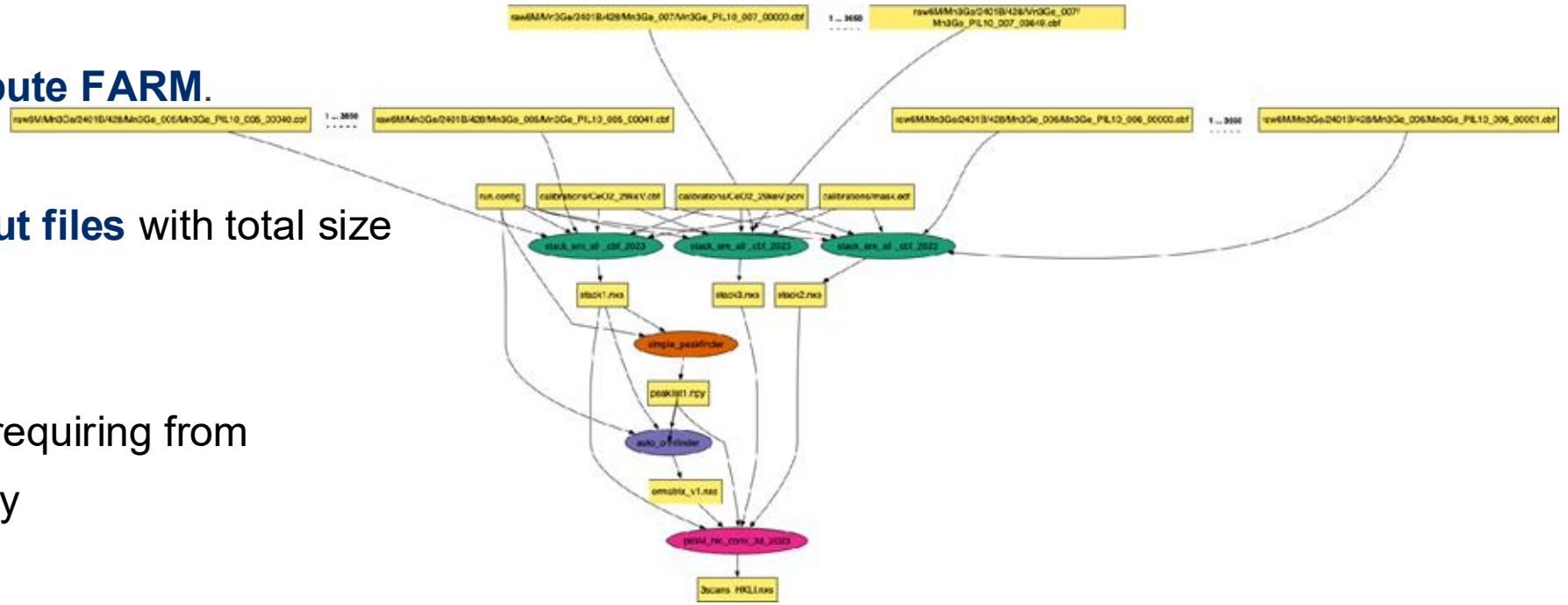
GitHub Repository: https://github.com/pegasus-isi/chess-qmb-workflow

Runs on the **CHESS Compute FARM**.

Requires about **11,000 input files** with total size of approximately **570GB**.

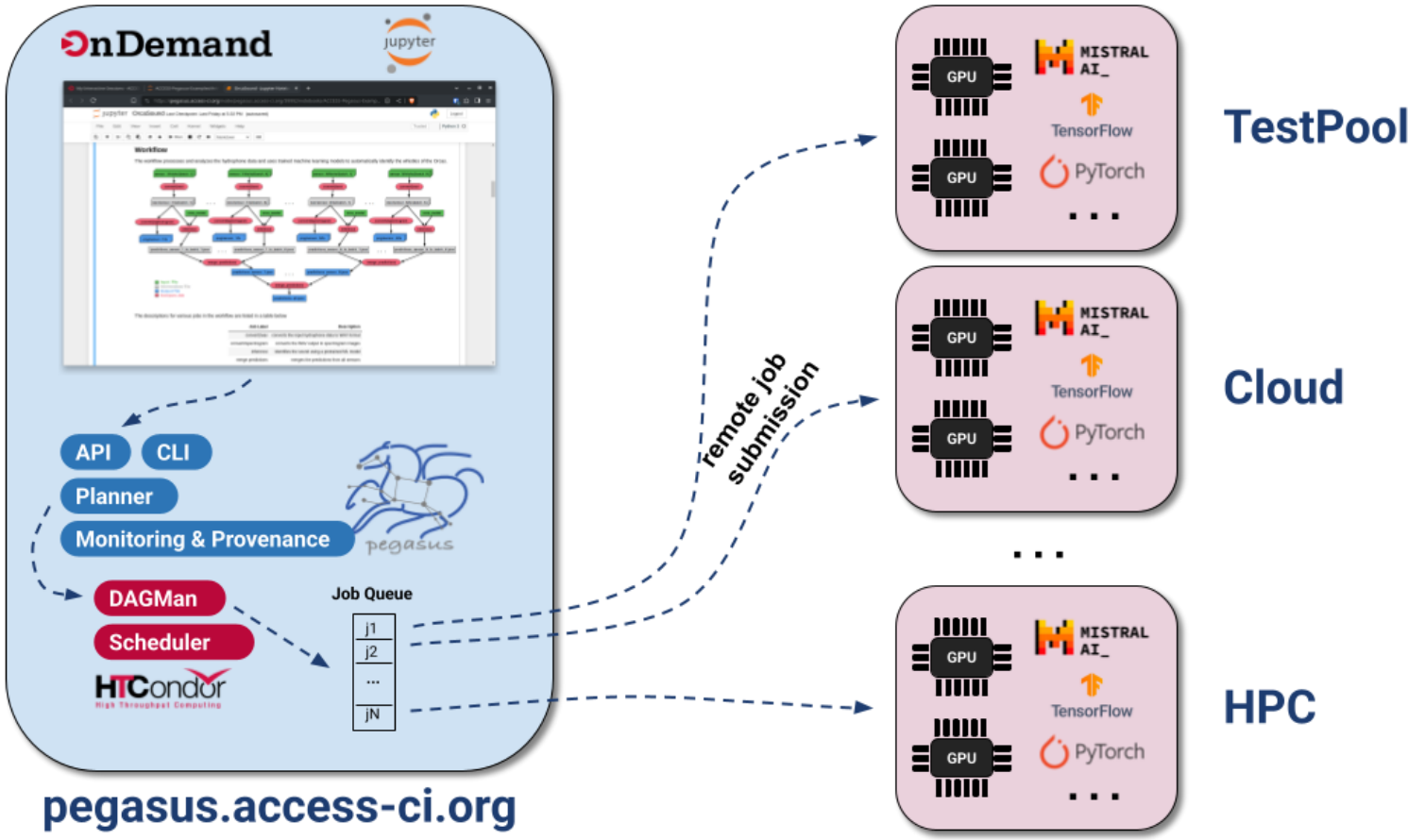
Mainly **high memory** jobs requiring from **10GB to 350 GB** of memory

The **pil6M_hkl_conv** job requries 56 cores.



Funded by the National Science Foundation, Grant #2320373

ACCESS Pegasus: Remote Job Submission



Remote job submission across resources

ACCESS Pegasus (pegasus.access-ci.org) plans each workflow and dispatches jobs through HTCondor/DAGMan to TestPool, Cloud, and HPC execution sites.

renci
USC Viterbi School of Engineering Information Sciences Institute
CHESS CORNELL HIGH ENERGY SYNCHROTRON SOURCE

Funded by the National Science Foundation, Grant #2320373

Getting Started with ACCESS Pegasus

1. **Get an ACCESS account** — register at https://operations.access-ci.org/identity/new-user
2. **Get an allocation (optional)** — not required for the training notebooks; start with an EXPLORE allocation for production work (see the DC200 module)
3. **Try Pegasus** — visit https://support.access-ci.org/tools/pegasus and log in with your ACCESS ID
4. **Run the examples** — open a Jupyter notebook and work through the ACCESS-Pegasus-Examples directory (01-Quickstart through 09-Tutorial-SharedFS)

Get support:

- Email: pegasus-support@isi.edu
- Slack: pegasus-users.slack.com
- ACCESS Support Portal: https://support.access-ci.org

Funded by the National Science Foundation, Grant #2320373

Galaxy: Web-Based Analysis Gateway

- Open-source, browser-based gateway — originally for bioinformatics, now used across many domains including X-ray science, genomics, and climate modeling
- Build, run, and share complex data-analysis workflows entirely in a browser — no programming or command-line experience required (use public usegalaxy.org or a local instance)
- **Key features:**
  - **Tool Integration** — a large, admin-managed library of analysis tools, with no software installs or version conflicts
  - **Workflow Builder** — a visual editor that connects tools on a canvas, encoding data dependencies automatically
  - **History** — records every dataset, tool, and parameter as a shareable, exportable audit trail
  - **Data Import & ToolShed** — import from uploads, URLs, shared libraries, and services like NCBI and Globus; reuse community tools via the Galaxy ToolShed
- **At CHESS:** a Galaxy instance configured for X-ray data analysis that submits jobs to the CHESS SGE cluster

Funded by the National Science Foundation, Grant #2320373

Summary

- Science gateways — Open OnDemand, Galaxy, and ACCESS Pegasus — sharply lower the barriers to using advanced computing resources
- They span everything from interactive analysis in a Jupyter notebook on a national HPC cluster to orchestrating complex, multi-step workflows across distributed resources — with no deep HPC system-administration expertise required
- **For CHESS researchers:**
  - **Open OnDemand** — the easiest way to get started on ACCESS HPC: open a browser, log in, and launch a Jupyter notebook on the cluster
  - **ACCESS Pegasus** — the best path for running Pegasus workflow pipelines on national CI without the setup overhead
  - Both work seamlessly with the ACCESS allocation described in the DC200 module

Thank you!

XCITE Gateways Module: https://xcitecourse.org/theme3/DC102/using-science-gateways.html

XCITE Workflows Module: https://xcitecourse.org/theme3/DC101/scientific-workflow-management.html

Pegasus Website: https://pegasus.isi.edu

National Science Foundation

NSF Cybertraining Award # 2320373

renci USC Viterbi School of Engineering Information Sciences Institute CHESS Cornell High Energy Synchrotron Source