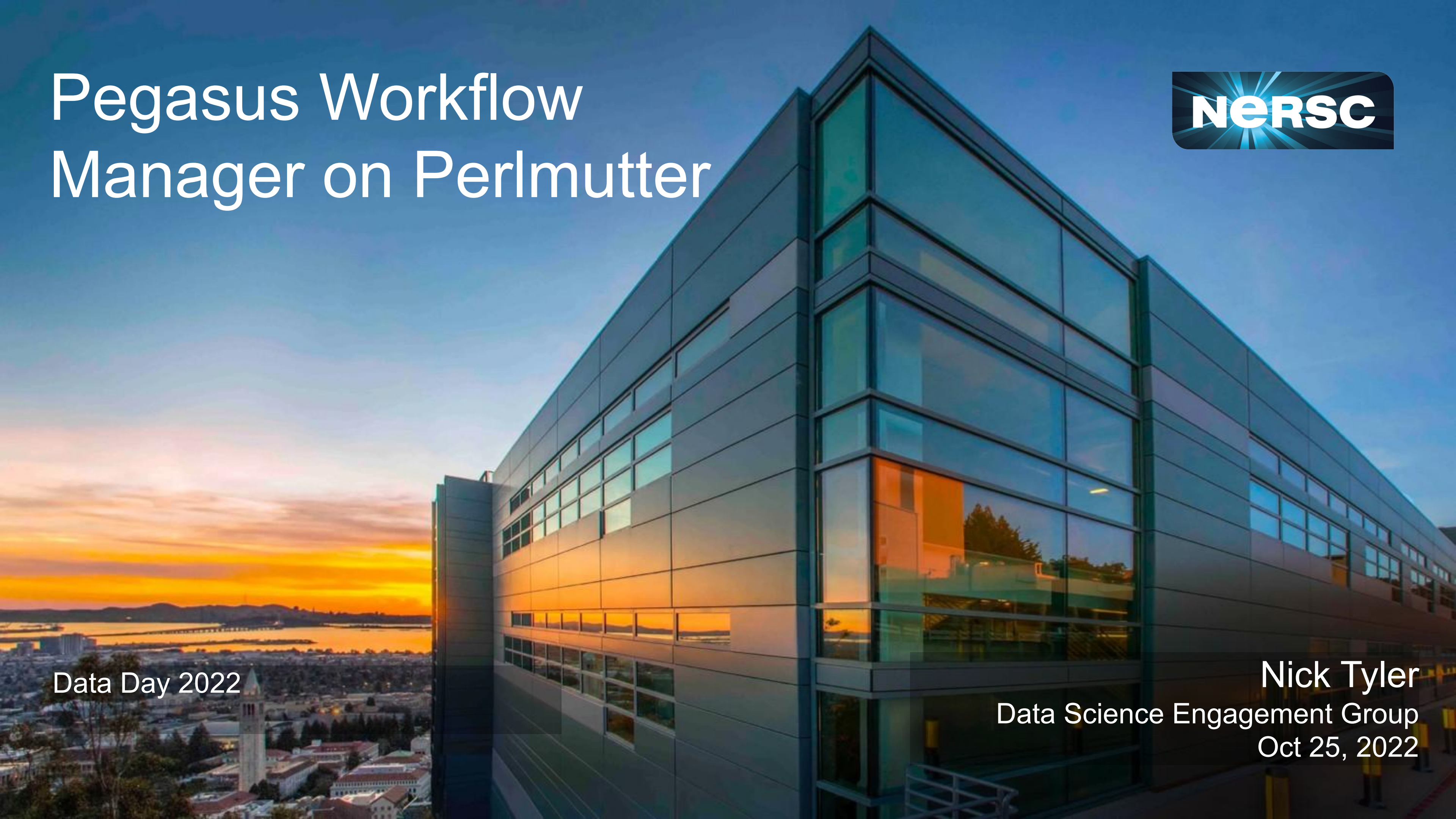


Pegasus Workflow Manager on Perlmutter

Data Day 2022

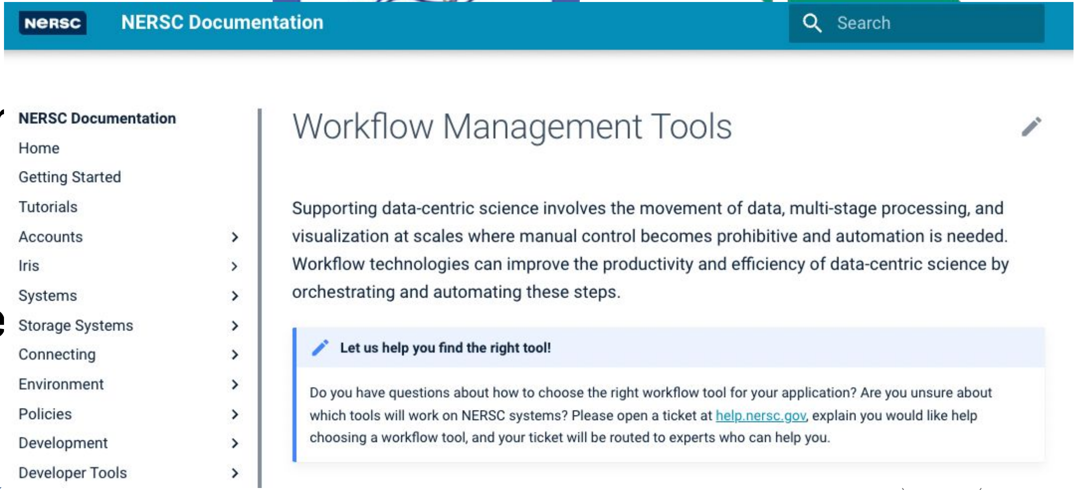
Nick Tyler
Data Science Engagement Group
Oct 25, 2022

What is a Workflow?

- Term “Workflow” is used a lot
- Different parts of analysis
- Steps in a data intensive process
    - Sbatch job script
    - Custom infrastructure
- Workflow Tools and Engines
    - Pegasus
    - Parsl/FuncX
    - Snakemake
    - Many more!

What is a Workflow?

- Term "Workflow" is used a lot
- Different parts of analysis
- Steps in a data intensive pr
  - Sbatch job script
  - Custom infrastructure
- Workflow Tools and Engine
  - Pegasus
  - Parsl/FuncX
  - Snakemake
  - [Many more!]()



Workflow Management Tools

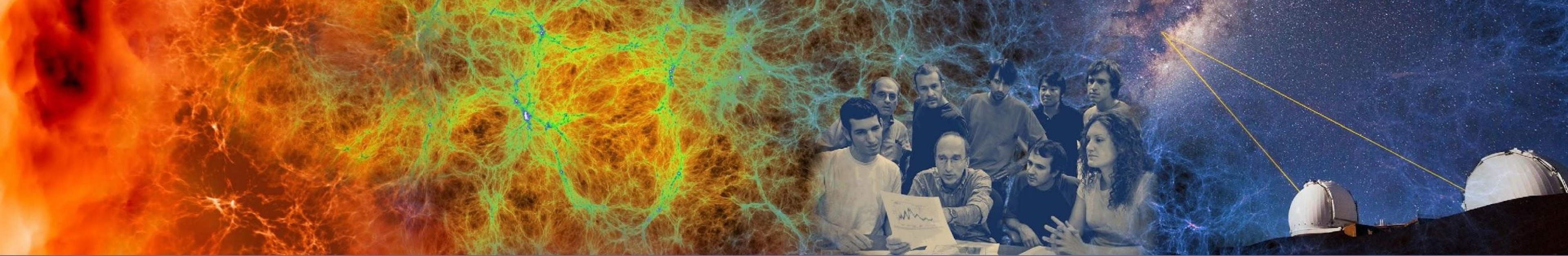
Supporting data-centric science involves the movement of data, multi-stage processing, and visualization at scales where manual control becomes prohibitive and automation is needed. Workflow technologies can improve the productivity and efficiency of data-centric science by orchestrating and automating these steps.

Let us help you find the right tool!

Do you have questions about how to choose the right workflow tool for your application? Are you unsure about which tools will work on NERSC systems? Please open a ticket at help.nersc.gov, explain you would like help choosing a workflow tool, and your ticket will be routed to experts who can help you.

Goals of workflow tools

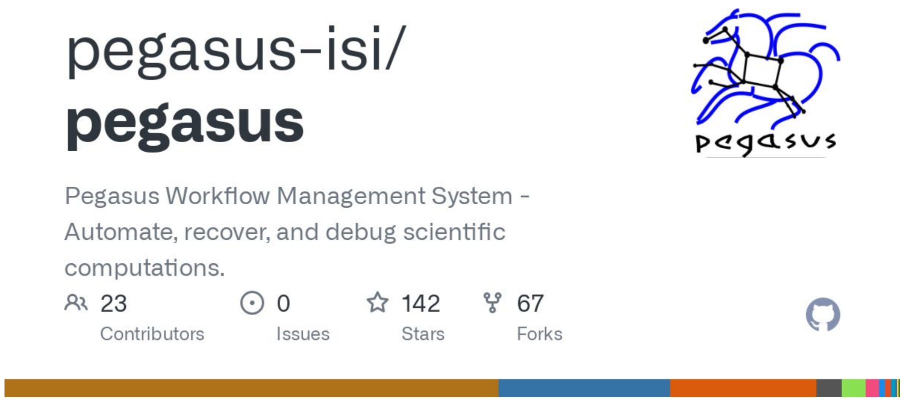
- Automation
- Reproducibility
- Share work with others
- Processed newest data
- Track data in the pipeline
- Use resources efficiently
- Get results faster
- Share work with others
- Less human in the loop



Pegasus Workflow Manager

What is pegasus?

- Workflow manager
- Define workflow using yaml files
  - replicas.yml
  - sites.yml
  - transformations.yml
  - workflow.yml
- There are APIs to create these yml files
  - Python, Java, R
- Show using Python API
- Example on data day github for perlmutter

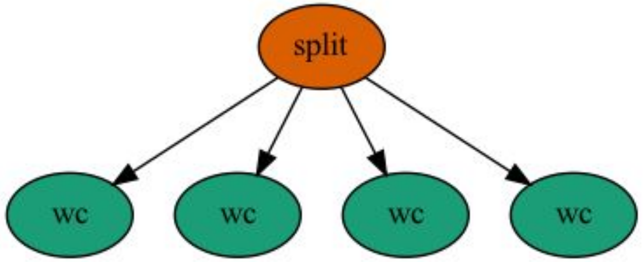

pegasus-isi/
pegasus

Pegasus Workflow Management System - Automate, recover, and debug scientific computations.

23 Contributors · 0 Issues · 142 Stars · 67 Forks

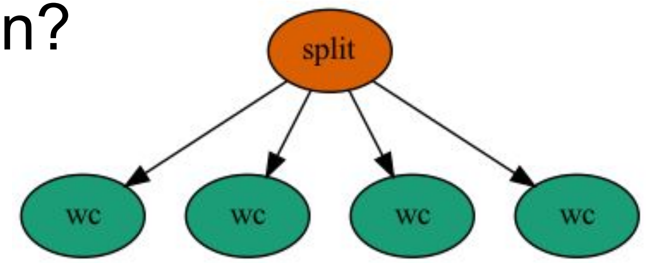
What is pegasus?

- Pegasus WMS
  - Pegasus APIs plan DAG
- Directed Acyclic Graph
  - Graph representing the work to be done
  - Nodes are executions
  - Edges show dataflow
  - Dependencies
- HTCondor Scheduler
  - DAGMan
  - Handling DAG execution
  - Scheduler manages execution of workflow



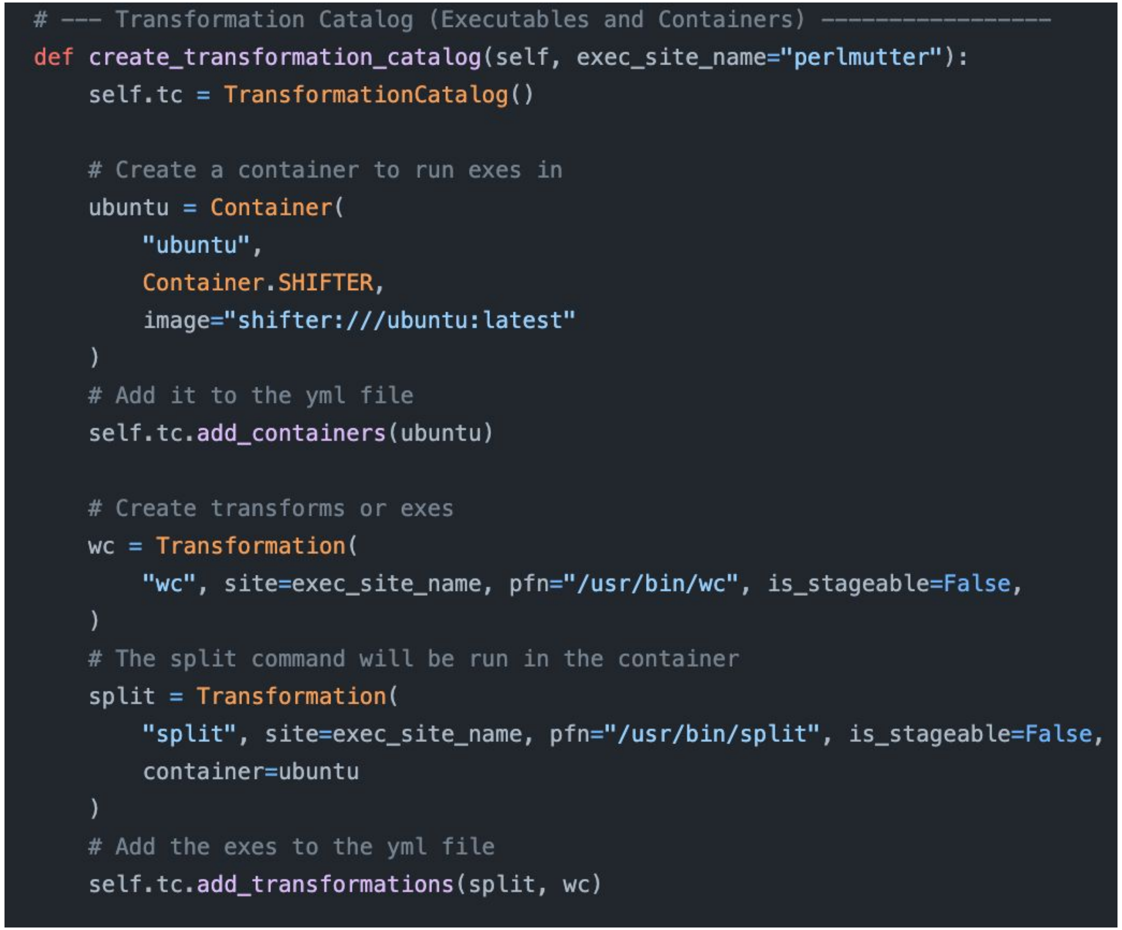
Writing the workflow

- Things to consider when building the workflow
- What executables are we going to run?
    - Are we using a container?
- What data do we have?
    - What are the inputs?
    - What are the outputs?
- What are the dependencies?
    - What tasks depend on outputs from previous tasks?
    - How are they connected?



Transforms

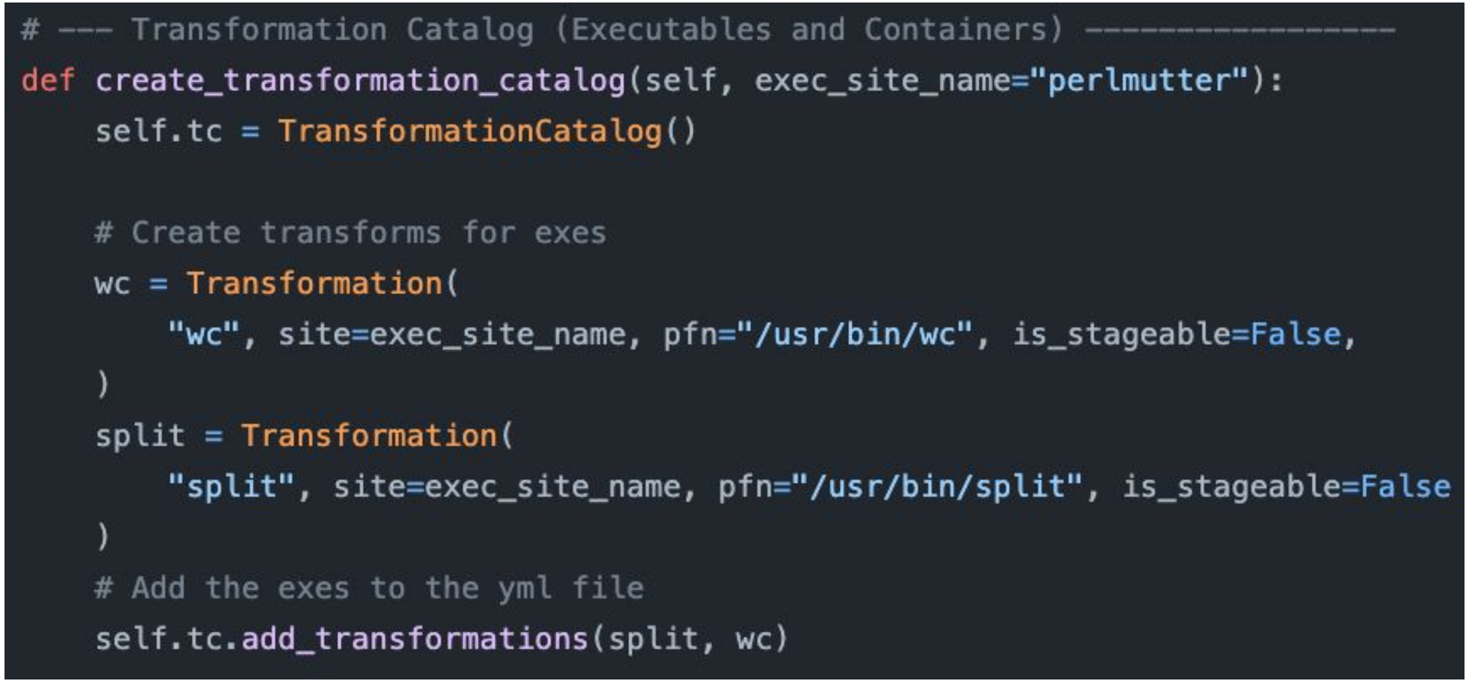
- Executables
- Containers



```python
# --- Transformation Catalog (Executables and Containers) -----------------
def create_transformation_catalog(self, exec_site_name="perlmutter"):
    self.tc = TransformationCatalog()

    # Create a container to run exes in
    ubuntu = Container(
        "ubuntu",
        Container.SHIFTER,
        image="shifter:///ubuntu:latest"
    )
    # Add it to the yml file
    self.tc.add_containers(ubuntu)

    # Create transforms or exes
    wc = Transformation(
        "wc", site=exec_site_name, pfn="/usr/bin/wc", is_stageable=False,
    )
    # The split command will be run in the container
    split = Transformation(
        "split", site=exec_site_name, pfn="/usr/bin/split", is_stageable=False,
        container=ubuntu
    )
    # Add the exes to the yml file
    self.tc.add_transformations(split, wc)
```

Transforms

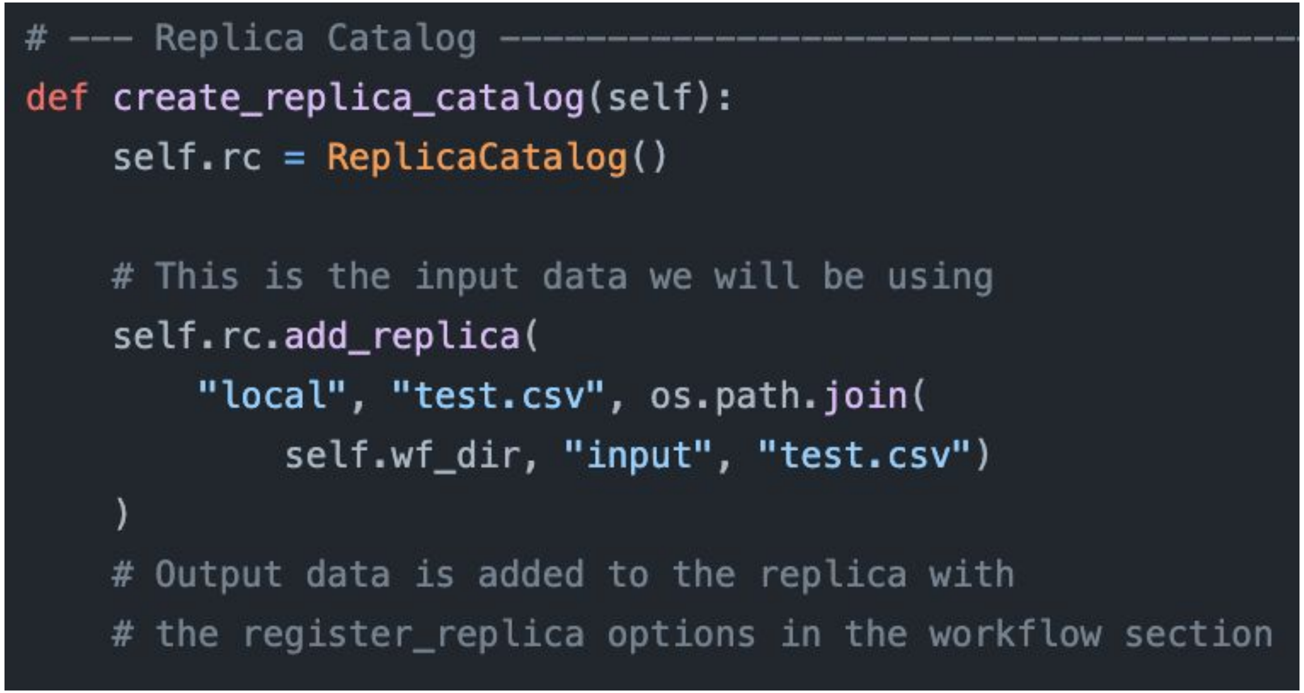
- Define Executables



```python
# --- Transformation Catalog (Executables and Containers) -----------------
def create_transformation_catalog(self, exec_site_name="perlmutter"):
    self.tc = TransformationCatalog()

    # Create transforms for exes
    wc = Transformation(
        "wc", site=exec_site_name, pfn="/usr/bin/wc", is_stageable=False,
    )
    split = Transformation(
        "split", site=exec_site_name, pfn="/usr/bin/split", is_stageable=False
    )
    # Add the exes to the yml file
    self.tc.add_transformations(split, wc)
```

Replica

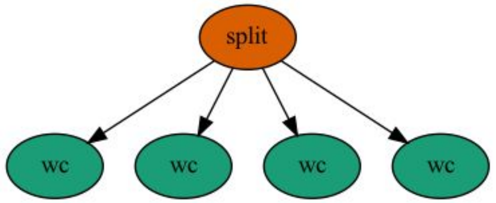
- Defines Data

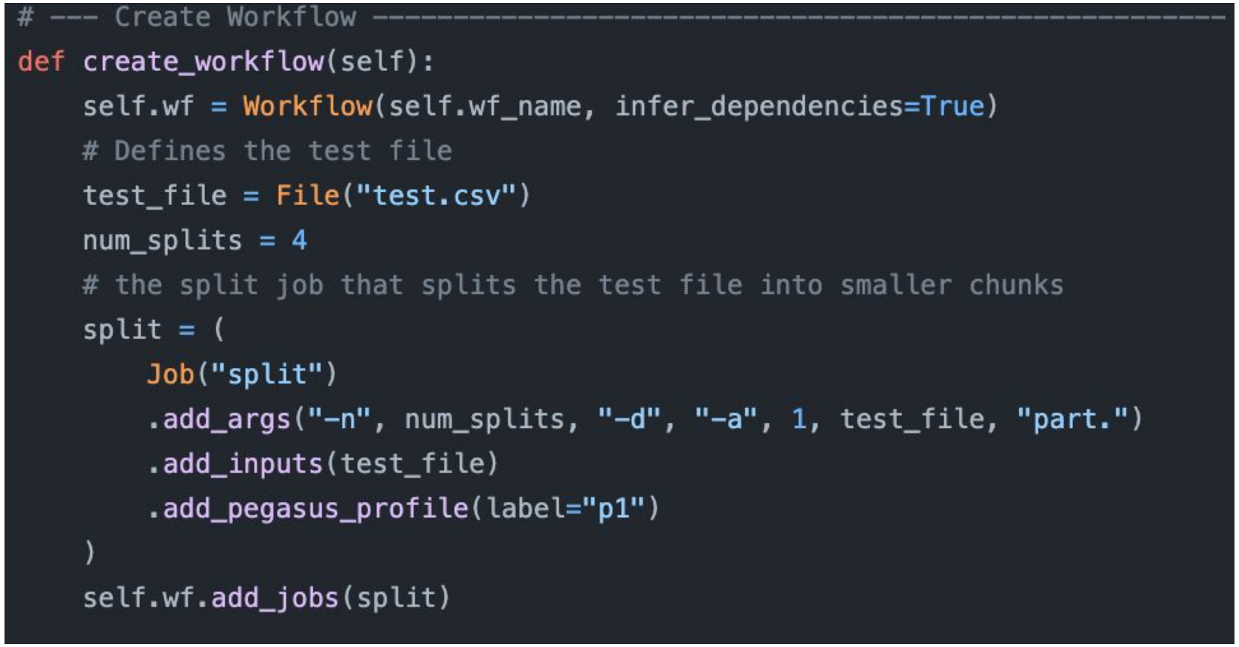


```
# --- Replica Catalog ------------------------------------
def create_replica_catalog(self):
    self.rc = ReplicaCatalog()

    # This is the input data we will be using
    self.rc.add_replica(
        "local", "test.csv", os.path.join(
            self.wf_dir, "input", "test.csv")
    )
    # Output data is added to the replica with
    # the register_replica options in the workflow section
```

Workflow

- Using the transforms and replicas let's build the workflow

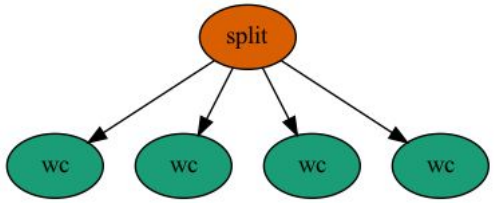


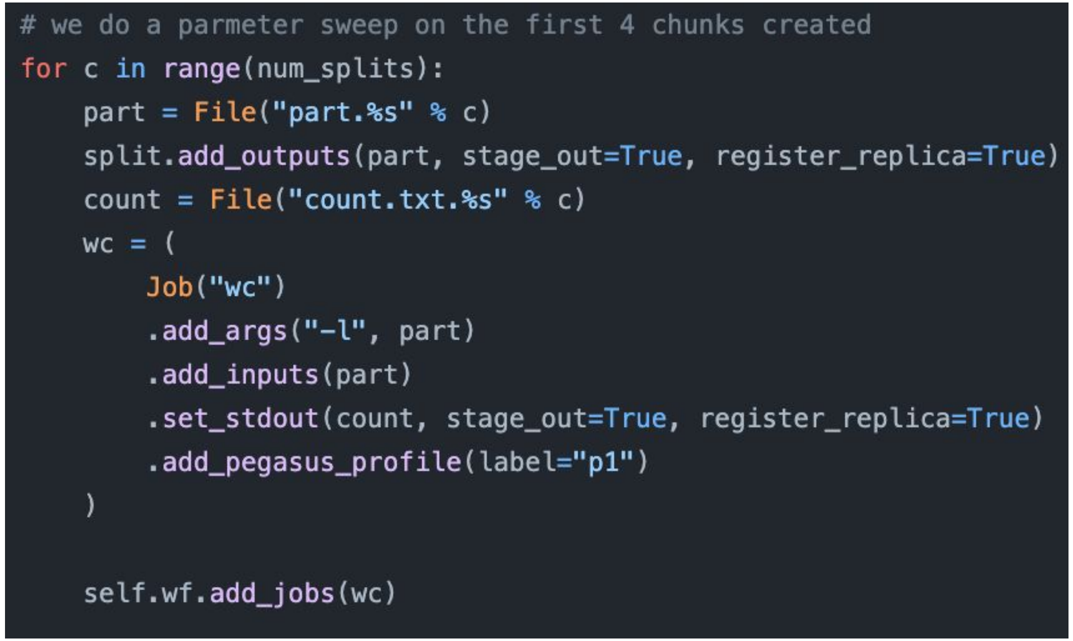


```
# --- Create Workflow -----------------------------------------------------
def create_workflow(self):
    self.wf = Workflow(self.wf_name, infer_dependencies=True)
    # Defines the test file
    test_file = File("test.csv")
    num_splits = 4
    # the split job that splits the test file into smaller chunks
    split = (
        Job("split")
        .add_args("-n", num_splits, "-d", "-a", 1, test_file, "part.")
        .add_inputs(test_file)
        .add_pegasus_profile(label="p1")
    )
    self.wf.add_jobs(split)
```

Workflow

- Using the transforms and replicas let's build the workflow





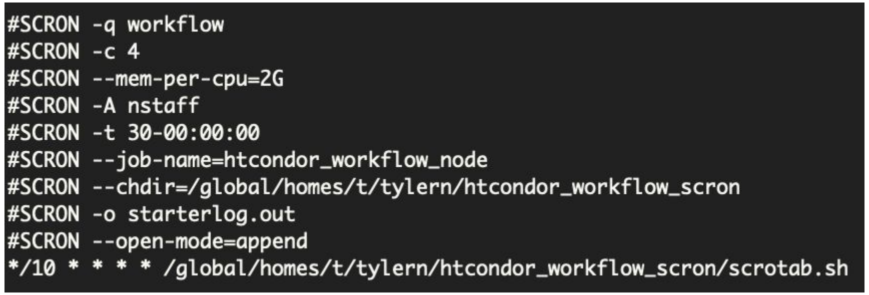
```
# we do a parmeter sweep on the first 4 chunks created
for c in range(num_splits):
    part = File("part.%s" % c)
    split.add_outputs(part, stage_out=True, register_replica=True)
    count = File("count.txt.%s" % c)
    wc = (
        Job("wc")
        .add_args("-l", part)
        .add_inputs(part)
        .set_stdout(count, stage_out=True, register_replica=True)
        .add_pegasus_profile(label="p1")
    )

    self.wf.add_jobs(wc)
```

Generating the workflow

- `python generate_workflow.py`
  - Creates the yml files
- replicas.yml
  - Defines the storage and data
- sites.yml
  - Defines the job parameters for the site sbatch
- transformations.yml
  - Defines the executables and their parameters
- workflow.yml
  - Defines the workflow
  - Workflows use transformations of replicas

Getting setup to run your workflow

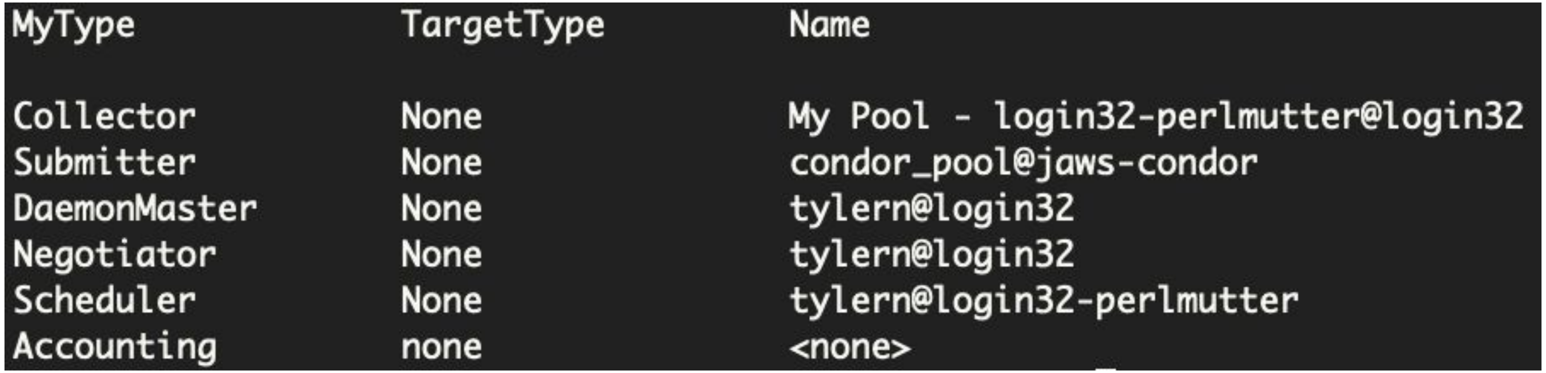
- For running on perlmutter we'll setup HTCondor as a workflow job
  - Uses scrontab to setup longer running workflow jobs
- HTCondor is a job scheduler
  - Built for High Throughput workloads
  - 100s-1000s of small jobs
  - Small job requirements (<< 1Node)
- Pegasus uses HTCondor to run workflows
  - HTCondor is a scheduler



```
#SCRON -q workflow
#SCRON -c 4
#SCRON --mem-per-cpu=2G
#SCRON -A nstaff
#SCRON -t 30-00:00:00
#SCRON --job-name=htcondor_workflow_node
#SCRON --chdir=/global/homes/t/tylern/htcondor_workflow_scron
#SCRON -o starterlog.out
#SCRON --open-mode=append
*/10 * * * * /global/homes/t/tylern/htcondor_workflow_scron/scrotab.sh
```

Getting setup to run your workflow

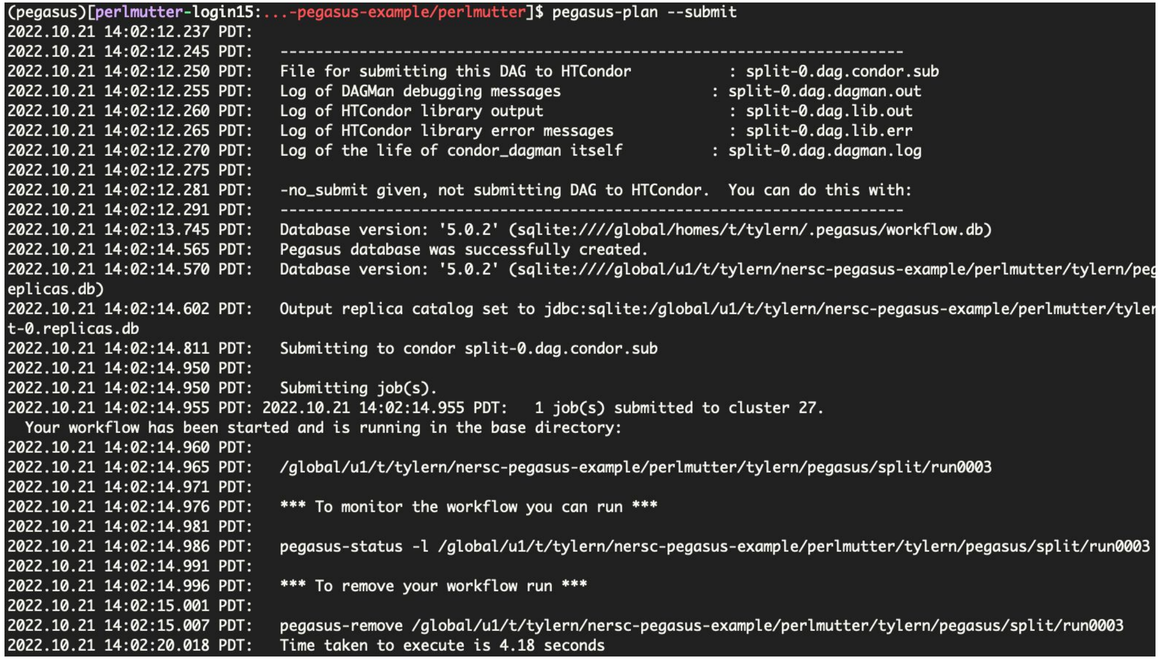
- Check that we have HTCondor working
- `condor_status -any`



```
MyType          TargetType      Name

Collector       None            My Pool - login32-perlmutter@login32
Submitter       None            condor_pool@jaws-condor
DaemonMaster    None            tylern@login32
Negotiator      None            tylern@login32
Scheduler       None            tylern@login32-perlmutter
Accounting      none            <none>
```

Pegasus Commands

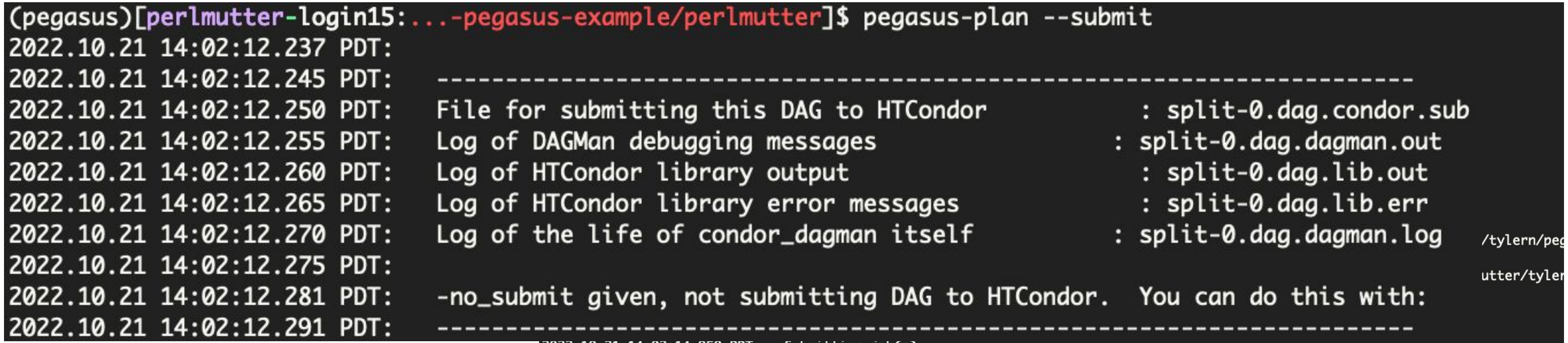
- `pegasus-plan --submit`



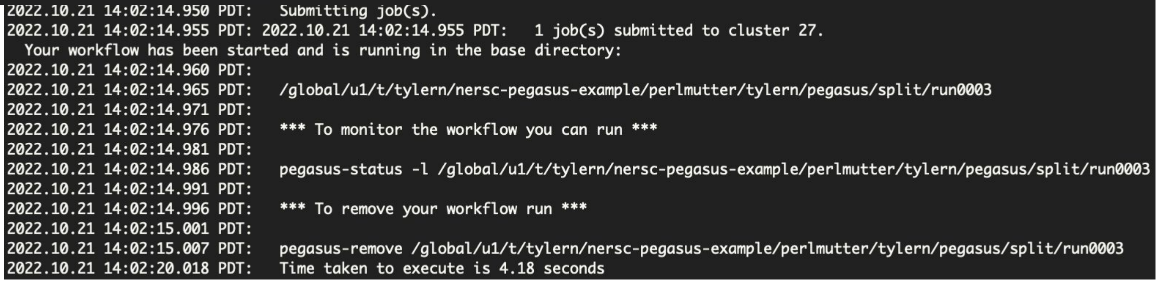
```
(pegasus)[perlmutter-login15:...-pegasus-example/perlmutter]$ pegasus-plan --submit
2022.10.21 14:02:12.237 PDT:
2022.10.21 14:02:12.245 PDT:   -----------------------------------------------------------------------
2022.10.21 14:02:12.250 PDT:   File for submitting this DAG to HTCondor           : split-0.dag.condor.sub
2022.10.21 14:02:12.255 PDT:   Log of DAGMan debugging messages                 : split-0.dag.dagman.out
2022.10.21 14:02:12.260 PDT:   Log of HTCondor library output                     : split-0.dag.lib.out
2022.10.21 14:02:12.265 PDT:   Log of HTCondor library error messages             : split-0.dag.lib.err
2022.10.21 14:02:12.270 PDT:   Log of the life of condor_dagman itself          : split-0.dag.dagman.log
2022.10.21 14:02:12.275 PDT:
2022.10.21 14:02:12.281 PDT:   -no_submit given, not submitting DAG to HTCondor.  You can do this with:
2022.10.21 14:02:12.291 PDT:   -----------------------------------------------------------------------
2022.10.21 14:02:13.745 PDT:   Database version: '5.0.2' (sqlite:////global/homes/t/tylern/.pegasus/workflow.db)
2022.10.21 14:02:14.565 PDT:   Pegasus database was successfully created.
2022.10.21 14:02:14.570 PDT:   Database version: '5.0.2' (sqlite:////global/u1/t/tylern/nersc-pegasus-example/perlmutter/tylern/peg
eplicas.db)
2022.10.21 14:02:14.602 PDT:   Output replica catalog set to jdbc:sqlite:/global/u1/t/tylern/nersc-pegasus-example/perlmutter/tyler
t-0.replicas.db
2022.10.21 14:02:14.811 PDT:   Submitting to condor split-0.dag.condor.sub
2022.10.21 14:02:14.950 PDT:
2022.10.21 14:02:14.950 PDT:   Submitting job(s).
2022.10.21 14:02:14.955 PDT: 2022.10.21 14:02:14.955 PDT:   1 job(s) submitted to cluster 27.
  Your workflow has been started and is running in the base directory:
2022.10.21 14:02:14.960 PDT:
2022.10.21 14:02:14.965 PDT:   /global/u1/t/tylern/nersc-pegasus-example/perlmutter/tylern/pegasus/split/run0003
2022.10.21 14:02:14.971 PDT:
2022.10.21 14:02:14.976 PDT:   *** To monitor the workflow you can run ***
2022.10.21 14:02:14.981 PDT:
2022.10.21 14:02:14.986 PDT:   pegasus-status -l /global/u1/t/tylern/nersc-pegasus-example/perlmutter/tylern/pegasus/split/run0003
2022.10.21 14:02:14.991 PDT:
2022.10.21 14:02:14.996 PDT:   *** To remove your workflow run ***
2022.10.21 14:02:15.001 PDT:
2022.10.21 14:02:15.007 PDT:   pegasus-remove /global/u1/t/tylern/nersc-pegasus-example/perlmutter/tylern/pegasus/split/run0003
2022.10.21 14:02:20.018 PDT:   Time taken to execute is 4.18 seconds
```

Pegasus Commands

- `pegasus-plan --submit`



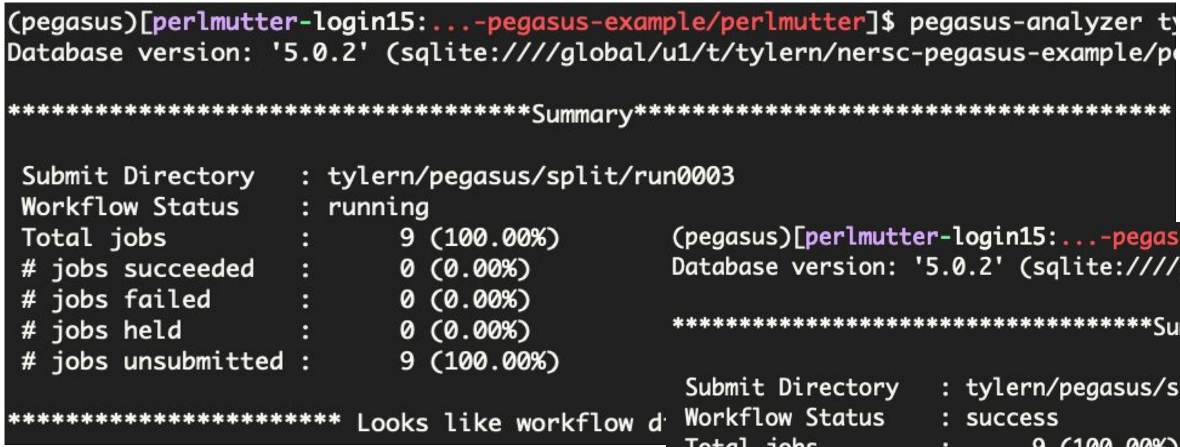
```
(pegasus)[perlmutter-login15:...-pegasus-example/perlmutter]$ pegasus-plan --submit
2022.10.21 14:02:12.237 PDT:
2022.10.21 14:02:12.245 PDT:   -----------------------------------------------------------------------
2022.10.21 14:02:12.250 PDT:   File for submitting this DAG to HTCondor           : split-0.dag.condor.sub
2022.10.21 14:02:12.255 PDT:   Log of DAGMan debugging messages                  : split-0.dag.dagman.out
2022.10.21 14:02:12.260 PDT:   Log of HTCondor library output                     : split-0.dag.lib.out
2022.10.21 14:02:12.265 PDT:   Log of HTCondor library error messages             : split-0.dag.lib.err
2022.10.21 14:02:12.270 PDT:   Log of the life of condor_dagman itself           : split-0.dag.dagman.log   /tylern/peg
2022.10.21 14:02:12.275 PDT:
                                                                                                          utter/tyler
2022.10.21 14:02:12.281 PDT:   -no_submit given, not submitting DAG to HTCondor.  You can do this with:
2022.10.21 14:02:12.291 PDT:   -----------------------------------------------------------------------
```



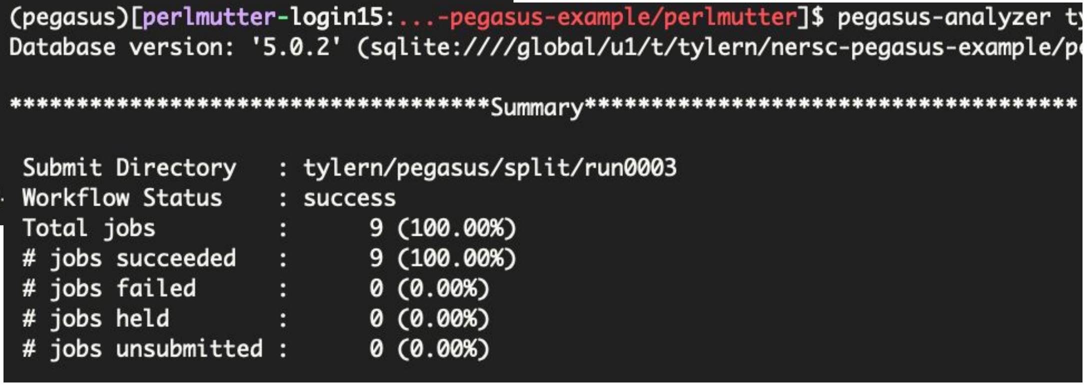
```
2022.10.21 14:02:14.950 PDT:   Submitting job(s).
2022.10.21 14:02:14.955 PDT: 2022.10.21 14:02:14.955 PDT:   1 job(s) submitted to cluster 27.
  Your workflow has been started and is running in the base directory:
2022.10.21 14:02:14.960 PDT:
2022.10.21 14:02:14.965 PDT:   /global/u1/t/tylern/nersc-pegasus-example/perlmutter/tylern/pegasus/split/run0003
2022.10.21 14:02:14.971 PDT:
2022.10.21 14:02:14.976 PDT:   *** To monitor the workflow you can run ***
2022.10.21 14:02:14.981 PDT:
2022.10.21 14:02:14.986 PDT:   pegasus-status -l /global/u1/t/tylern/nersc-pegasus-example/perlmutter/tylern/pegasus/split/run0003
2022.10.21 14:02:14.991 PDT:
2022.10.21 14:02:14.996 PDT:   *** To remove your workflow run ***
2022.10.21 14:02:15.001 PDT:
2022.10.21 14:02:15.007 PDT:   pegasus-remove /global/u1/t/tylern/nersc-pegasus-example/perlmutter/tylern/pegasus/split/run0003
2022.10.21 14:02:20.018 PDT:   Time taken to execute is 4.18 seconds
```

Pegasus Commands

- `pegasus-analyzer path/to/workflow/run0001`



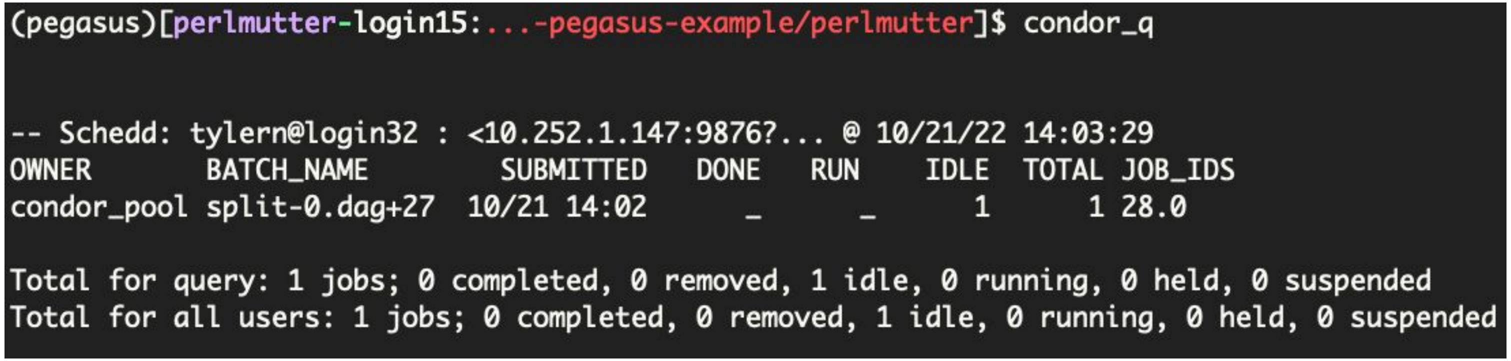
```
(pegasus)[perlmutter-login15:...-pegasus-example/perlmutter]$ pegasus-analyzer ty
Database version: '5.0.2' (sqlite:////global/u1/t/tylern/nersc-pegasus-example/pe

**********************************Summary***********************************

 Submit Directory   : tylern/pegasus/split/run0003
 Workflow Status    : running
 Total jobs         :      9 (100.00%)
 # jobs succeeded   :      0 (0.00%)
 # jobs failed      :      0 (0.00%)
 # jobs held        :      0 (0.00%)
 # jobs unsubmitted :      9 (100.00%)

*********************** Looks like workflow d
```



```
(pegasus)[perlmutter-login15:...-pegasus-example/perlmutter]$ pegasus-analyzer ty
Database version: '5.0.2' (sqlite:////global/u1/t/tylern/nersc-pegasus-example/pe

**********************************Summary***********************************

 Submit Directory   : tylern/pegasus/split/run0003
 Workflow Status    : success
 Total jobs         :      9 (100.00%)
 # jobs succeeded   :      9 (100.00%)
 # jobs failed      :      0 (0.00%)
 # jobs held        :      0 (0.00%)
 # jobs unsubmitted :      0 (0.00%)
```

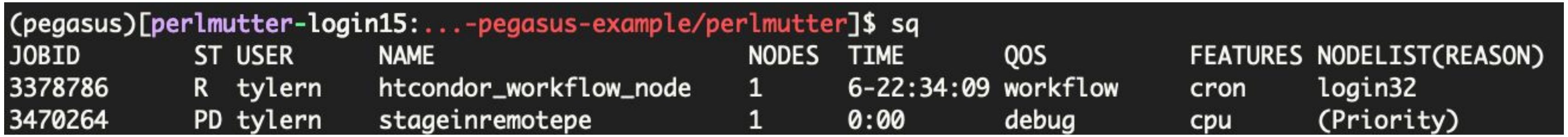
Watching the jobs progress

- condor_q



```
(pegasus)[perlmutter-login15:...-pegasus-example/perlmutter]$ condor_q


-- Schedd: tylern@login32 : <10.252.1.147:9876?... @ 10/21/22 14:03:29
OWNER        BATCH_NAME         SUBMITTED    DONE    RUN     IDLE   TOTAL JOB_IDS
condor_pool split-0.dag+27  10/21 14:02       _       _        1       1 28.0

Total for query: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
Total for all users: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
```

- sqs

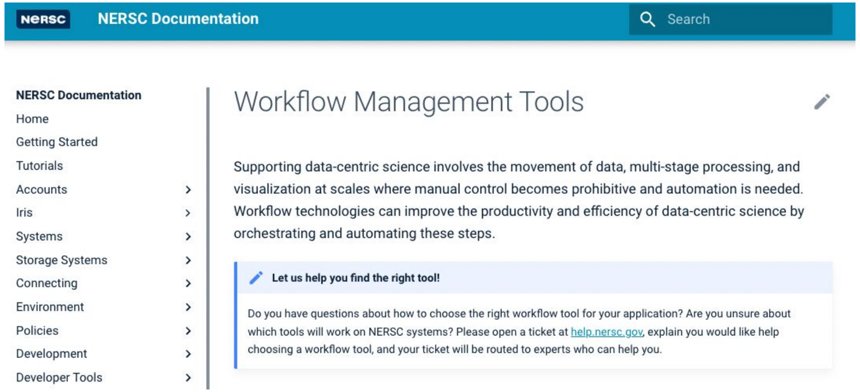


```
(pegasus)[perlmutter-login15:...-pegasus-example/perlmutter]$ sq
JOBID        ST USER      NAME                     NODES  TIME        QOS         FEATURES NODELIST(REASON)
3378786      R  tylern    htcondor_workflow_node   1      6-22:34:09 workflow     cron     login32
3470264      PD tylern    stageinremotepe          1      0:00        debug       cpu      (Priority)
```

Watching the jobs progress

Questions?

- Pegasus is just one of many workflow tools
- Each has its advantages and disadvantages
- Checkout all the tools we have on our docs



NERSC Documentation
Home
Getting Started
Tutorials
Accounts
Iris
Systems
Storage Systems
Connecting
Environment
Policies
Development
Developer Tools

Workflow Management Tools

Supporting data-centric science involves the movement of data, multi-stage processing, and visualization at scales where manual control becomes prohibitive and automation is needed. Workflow technologies can improve the productivity and efficiency of data-centric science by orchestrating and automating these steps.

Let us help you find the right tool!

Do you have questions about how to choose the right workflow tool for your application? Are you unsure about which tools will work on NERSC systems? Please open a ticket at help.nersc.gov, explain you would like help choosing a workflow tool, and your ticket will be routed to experts who can help you.