

# FireWorks



DOE Cross-facility workshop

April 12, 2023

Laurie Stephey (NERSC) and Sean Wilkinson (OLCF)

# Thanks!

Thanks to Anubhav Jain, FireWorks developer, for help and advice developing our demo

Thanks to Urjoshi Sinha and Daniel Fulton at NERSC, who created ~150 MongoDBs for training accounts + NERSC users

Thanks to Alvaro Vazquez at ALCF who helped provide instructions for creating MongoDBs at ALCF

# What is FireWorks?

- FireWorks is a general purpose workflow tool with a MongoDB backend
- Free and open source at <https://github.com/materialsproject/fireworks>
- Developed and maintained by The Materials Project at LBL
- Detailed documentation at <https://materialsproject.github.io/fireworks/>
- Active user community forum at <https://matsci.org/c/fireworks/15>
- Language agnostic (can run via CLI or Python API)

FireWorks

The logo consists of the word "FireWorks" in a large, bold, dark blue sans-serif font. A thin blue horizontal line starts from the top of the letter "F" and extends to the right, ending with a small rectangular box. From this box, several blue and purple lines radiate outwards, resembling fireworks or light rays.

# When to use FireWorks

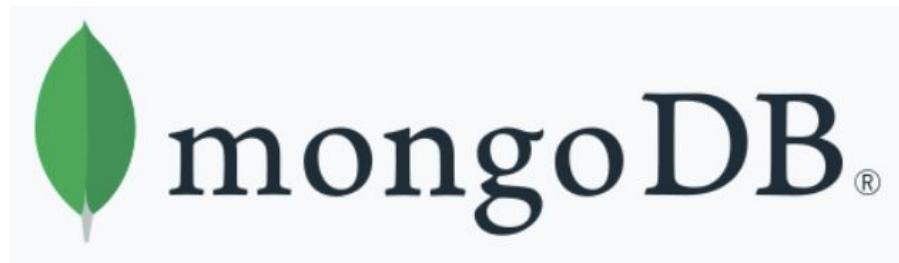
- **Resilience and workflow provenance:** MongoDB sits outside the job and can track and re-launch failed workflows
- **MPI:** FireWorks workflows can support **multi-node MPI tasks**
- **High-throughput:** FireWorks excels at high-throughput workflows, especially tasks with the same resource requirements
- **Web GUI:** FireWorks provides a way to monitor and inspect the state of workflows using your web browser

# When not to use FireWorks

- **Quick setup:** if you want to get up and running quickly and don't want to manage a MongoDB
- **Short term:** if you don't need workflow information to persist beyond the lifetime of a job, FireWorks may be overkill
- **Heterogeneous workflows:** If your workflow includes many different sized tasks, FireWorks can be cumbersome
- **1:1 correspondence between jobs launched and jobs that will run your workflow**
  - FireWorks may launch more jobs than steps in your workflow. Some jobs will wake up and quit if their dependencies are still pending. By default FireWorks retries once per minute.

# MongoDB - stores all information in FireWorks

- MongoDB is a NoSQL type database
- MongoDB is used to store information about workflow and job status
- Many users treat MongoDB as a long-lived service that may run on a separate platform
- Information in the MongoDB can be queried at the command line or visualized using a dashboard (see Extra Slides)



# Setting up a MongoDB

- At ALCF, users can set up a MongoDB inside a screen process on a Polaris login node following [directions](#) here
- At NERSC, our database team will create a MongoDB for you upon request
  - [Directions](#) for existing NERSC accounts, created for users who requested
  - [Directions](#) for NERSC training accounts, created for all training accounts
  - Long term databases can be requested using our [form](#)
- At OLCF, users can provision their own MongoDB on Slate, a set of OpenShift (Kubernetes) clusters
  - [https://docs.olcf.ornl.gov/services\\_and\\_applications/slate/use\\_cases/mongodb\\_service.html](https://docs.olcf.ornl.gov/services_and_applications/slate/use_cases/mongodb_service.html)

# Installing FireWorks

- [Directions for installing FireWorks at ALCF](#)
- [Directions for installing FireWorks at NERSC](#)
- [Directions for installing FireWorks at OLCF](#)

# FireWorks 101

- **LaunchPad**- `my_launchpad.yaml`, used to collect FireTasks/FireWorks that are ready to run, interfaces with the MongoDB
- **FireWork**- `fw_diabetes_ht.yaml`, file that describes a workflow (several FireTasks)
- **FireWorker**- `my_fworker.yaml`, configuration file for resource that will run your job
- **QueueAdapter**- `my_qadapter.yaml`, file that links FireWorks to the HPC center queue (different formats for Slurm, PBS, etc)
- **rlaunch**- runs the FireTask/FireWork on the FireServer (local node)
- **qlaunch**- submits a job to the HPC scheduler

# FireWorks QueueAdapter used to submit jobs

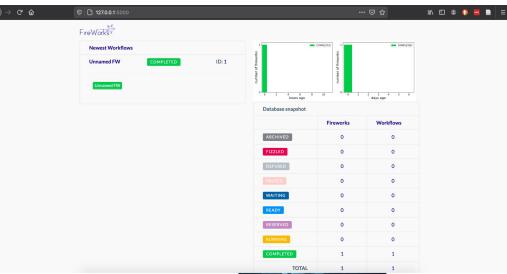
- FireWorks can submit jobs to a queue (most common use case for HPC)
- The “bridge” between FireWorks and your targeted HPC system is the `my_qadapter.yaml` file
- You can find `my_qadapter.yaml` templates for many different schedulers, including PBS and Slurm in the [FireWorks Github repo examples](#)
- This is the file where you can change job resource specifications (time limit, number of nodes, CPUs/GPUs, etc)
- You can have more than one `my_qadapter.yaml` if you need multiple types of resources, as we'll show in our demo

# Using FireWorks on an HPC system

Database stores FireWorks information



Dashboard can be used to display status in database (see Extra Slides)



A screenshot of the FireWorks dashboard. At the top, there are tabs for 'Newest Workflows' (highlighted), 'Unseen FW', and 'Completed'. Below the tabs is a 'Database snapshot' section showing two bar charts: 'Workflow Progress' and 'Workflow Status'. The 'Workflow Progress' chart shows a green bar at 100% and a red bar at 0%. The 'Workflow Status' chart shows a green bar at 100% and a red bar at 0%. Below these are sections for 'Fireworks' and 'Workflows', each with a table of status counts. A legend at the bottom defines colors for different workflow states: GREEN (COMPLETED, PENDING, SUBMITTED, REQUEUED, RESTARTED, UNKNOWN), RED (FAILED, CANCELLED), and GREY (WAITING). A footer at the bottom right says 'National Laboratory | FACILITY'.

Describe your workflow in in a *FireWork*: `fw.yaml`

Configure the *FireWorker* that will run your job  
`my_fworker.yaml`

Request your job resources in a *QueueAdapter*:  
`my_qadapter.yaml`

Add your *FireWork* workflow to the LaunchPad with  
`lpad add fw.yaml`

Use `qlaunch` to submit your *FireWork* workflow to the queue. You may also specify the *FireWorker* and *QueueAdapter*



# Our demo uses the scikit-learn diabetes dataset

The [diabetes dataset](#) comes built-in in the scikit-learn Python library

## Data Set Characteristics:

<b>Number of Instances:</b>	442
<b>Number of Attributes:</b>	First 10 columns are numeric predictive values
<b>Target:</b>	Column 11 is a quantitative measure of disease progression one year after baseline
<b>Attribute Information:</b>	<ul style="list-style-type: none"><li>• age age in years</li><li>• sex</li><li>• bmi body mass index</li><li>• bp average blood pressure</li><li>• s1 tc, total serum cholesterol</li><li>• s2 ldl, low-density lipoproteins</li><li>• s3 hdl, high-density lipoproteins</li><li>• s4 tch, total cholesterol / HDL</li><li>• s5 ltg, possibly log of serum triglycerides level</li><li>• s6 glu, blood sugar level</li></ul>

***Which of the 10 attributes  
is most highly correlated  
with the measure of  
diabetes disease  
progression? Let's run the  
demo to find out!***

# What is happening in our demo?

- **step\_1\_diabetes\_preprocessing.py**: we load the diabetes dataset, write it to two numpy .npy files
- **step\_2\_diabetes\_correlation.py**: we load the .npy files and process each attribute in a separate MPI rank across 2 nodes. We calculate the Pearson correlation coefficient between each attribute and the measure of disease progression. We gather the outputs back to a single file.
- **step\_3\_diabetes\_postprocessing.py**: we load the Pearson correlation coefficient data and pretty print it to the screen using Pandas

# Try it yourself!

- [Directions for running the demo at ALCF on Polaris](#)
- [Directions for running the demo at NERSC on Perlmutter](#)
- [Directions for running the demo at OLCF on Summit](#)

# Demo at NERSC with the CLI

We're going to do the harder Demo 2-  
heterogenous workflow demo

# Step 0- Set up environment

This assumes you already have set up your MongoDB, installed FireWorks, and configured your LaunchPad

```
[stephey@perlmutter:login26:~] module load python
[stephey@perlmutter:login26:~] conda activate fireworks
[(fireworks)stephey@perlmutter:login26:~] cd /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC
[(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> lpad res
et
[Are you sure? This will RESET 0 workflows and all data. (Y/N)y
2023-04-11 22:07:44,454 INFO Performing db tune-up
2023-04-11 22:07:44,518 INFO LaunchPad was RESET.
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC>
```

# Step 1- Let's look at our workflow FireWork file

```
[fireworks)stephey@login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> cat fw_diabetes_wf.yaml
fws:
- fw_id: 1
  spec:
    _category: onenode
    _launch_dir: /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC
    _tasks:
      - _fw_name: ScriptTask
        script: srun python step_1_diabetes_preprocessing.py
- fw_id: 2
  spec:
    _category: twonode
    _launch_dir: /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC
    _tasks:
      - _fw_name: ScriptTask
        script: srun -n 10 --cpu-bind=cores python step_2_diabetes_correlation.py
- fw_id: 3
  spec:
    _category: onenode
    _launch_dir: /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC
    _tasks:
      - _fw_name: ScriptTask
        script: srun python step_3_diabetes_postprocessing.py
links:
  1:
  - 2
  2:
  - 3
metadata: {}
(fireworks)stephey@login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC>
```

\_category is used to link the task to the FireWorker

# Step 2- Let's look at our FireWorker configuration files

```
[fireworks]stephey@login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> cat my_fworker1.yaml
name: one node fireworker
category: onenode
query: '{}'
[fireworks]stephey@login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> cat my_fworker2.yaml
name: two node fireworker
category: twonode
query: '{}'
[fireworks]stephey@login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC>
```

# Step 3- Let's look at our QueueAdapter files

```
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> cat my_qadapter1.yaml
_fw_name: CommonAdapter
_fw_q_type: SLURM
rocket_launch: rlaunch -w /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/my_fworker1.yaml -l /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/my_launchpad.yaml singleshot
constraint: cpu
nodes: 1
account: nstaff
walltime: '00:05:00'
queue: debug
job_name: null
logdir: null
pre_rocket: null
post_rocket: null
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> cat my_qadapter2.yaml
_fw_name: CommonAdapter
_fw_q_type: SLURM
rocket_launch: rlaunch -w /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/my_fworker2.yaml -l /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/my_launchpad.yaml singleshot
constraint: cpu
nodes: 2
account: nstaff
walltime: '00:05:00'
queue: debug
job_name: null
logdir: null
pre_rocket: null
post_rocket: null
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC>
```

# Step 4- Let's launch our heterogenous workflow

```
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> lpad reset
Are you sure? This will RESET 0 workflows and all data. (Y/N)y
2023-04-11 22:25:36,165 INFO Performing db tune-up
2023-04-11 22:25:36,224 INFO LaunchPad was RESET.
[1]+ Done qlaunch -q my_qadapter1.yaml rapidfire
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> lpad add fw_diabetes_wf.yaml
2023-04-11 22:25:41,657 INFO Added a workflow. id_map: {1: 1, 2: 2, 3: 3}
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> qlaunch -q my_qadapter1.yaml rapidfire & qlaunch -q my_qadapter2.yaml rapidfire
[1] 210839
2023-04-11 22:26:02,544 INFO getting queue adapter
2023-04-11 22:26:02,544 INFO Found previous block, using /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-04-45-27-649185
2023-04-11 22:26:02,545 INFO getting queue adapter
2023-04-11 22:26:02,545 INFO Found previous block, using /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-04-45-27-649185
2023-04-11 22:26:02,557 INFO Launching a rocket!
2023-04-11 22:26:02,557 INFO Launching a rocket!
2023-04-11 22:26:02,563 INFO Created new dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-04-45-27-649185/launcher_2023-04-12-05-26-02-562615
2023-04-11 22:26:02,563 INFO moving to launch_dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-04-45-27-649185/launcher_2023-04-12-05-26-02-562615
2023-04-11 22:26:02,563 INFO Created new dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-04-45-27-649185/launcher_2023-04-12-05-26-02-562916
2023-04-11 22:26:02,564 INFO moving to launch_dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-04-45-27-649185/launcher_2023-04-12-05-26-02-562916
2023-04-11 22:26:02,565 INFO submitting queue script
2023-04-11 22:26:02,565 INFO submitting queue script
2023-04-11 22:26:02,637 INFO Job submission was successful and job_id is 7226154
2023-04-11 22:26:02,638 INFO Sleeping for 5 seconds....zzz...
2023-04-11 22:26:02,649 INFO Job submission was successful and job_id is 7226155
2023-04-11 22:26:02,649 INFO Sleeping for 5 seconds....zzz...
2023-04-11 22:26:07,648 INFO Launching a rocket!
2023-04-11 22:26:07,654 INFO Created new dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-04-45-27-649185/launcher_2023-04-12-05-26-07-653551
```

# Step 5- Check status in another window

```
[(fireworks)stephey@perlmutter:login10:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/Fireworks/NERSC> lpad get_fws
[
  {
    "fw_id": 1,
    "created_on": "2023-04-12T05:29:45.040624",
    "updated_on": "2023-04-12T05:29:52.005459",
    "state": "RUNNING",
    "name": "Unnamed FW"
  },
  {
    "fw_id": 2,
    "created_on": "2023-04-12T05:29:45.040740",
    "updated_on": "2023-04-12T05:29:45.040740",
    "name": "Unnamed FW",
    "state": "WAITING"
  },
  {
    "fw_id": 3,
    "created_on": "2023-04-12T05:29:45.040814",
    "updated_on": "2023-04-12T05:29:45.040814",
    "name": "Unnamed FW",
    "state": "WAITING"
  }
]
(fireworks)stephey@perlmutter:login10:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/Fireworks/NERSC>
```

# Step 6- Workflow is done!

```
2023-04-11 22:27:12,900 INFO submitting queue script
2023-04-11 22:27:12,900 INFO submitting queue script
2023-04-11 22:27:12,978 INFO Job submission was successful and job_id is 7226172
2023-04-11 22:27:12,978 INFO Sleeping for 5 seconds...zzz...
2023-04-11 22:27:12,986 INFO Job submission was successful and job_id is 7226173
2023-04-11 22:27:12,986 INFO Sleeping for 5 seconds...zzz...
2023-04-11 22:27:17,988 INFO Launching a rocket!
2023-04-11 22:27:17,993 INFO Created new dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-45-27-649185/launcher_2023-04-12-05-27-17-992595
2023-04-11 22:27:17,993 INFO moving to launch_dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-45-27-649185/launcher_2023-04-12-05-27-17-992595
2023-04-11 22:27:17,995 INFO submitting queue script
2023-04-11 22:27:17,996 INFO Launching a rocket!
2023-04-11 22:27:18,000 INFO Created new dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-45-27-649185/launcher_2023-04-12-05-27-18-000215
2023-04-11 22:27:18,000 INFO moving to launch_dir /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-10-45-27-649185/launcher_2023-04-12-05-27-18-000215
2023-04-11 22:27:18,002 INFO submitting queue script
2023-04-11 22:27:18,068 INFO Job submission was successful and job_id is 7226175
2023-04-11 22:27:18,068 INFO Sleeping for 5 seconds...zzz...
2023-04-11 22:27:18,070 INFO Job submission was successful and job_id is 7226176
2023-04-11 22:27:18,071 INFO Sleeping for 5 seconds...zzz...
[1]+ Done                  qlaunch -q my_qadapter1.yaml rapidfire
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC>
```

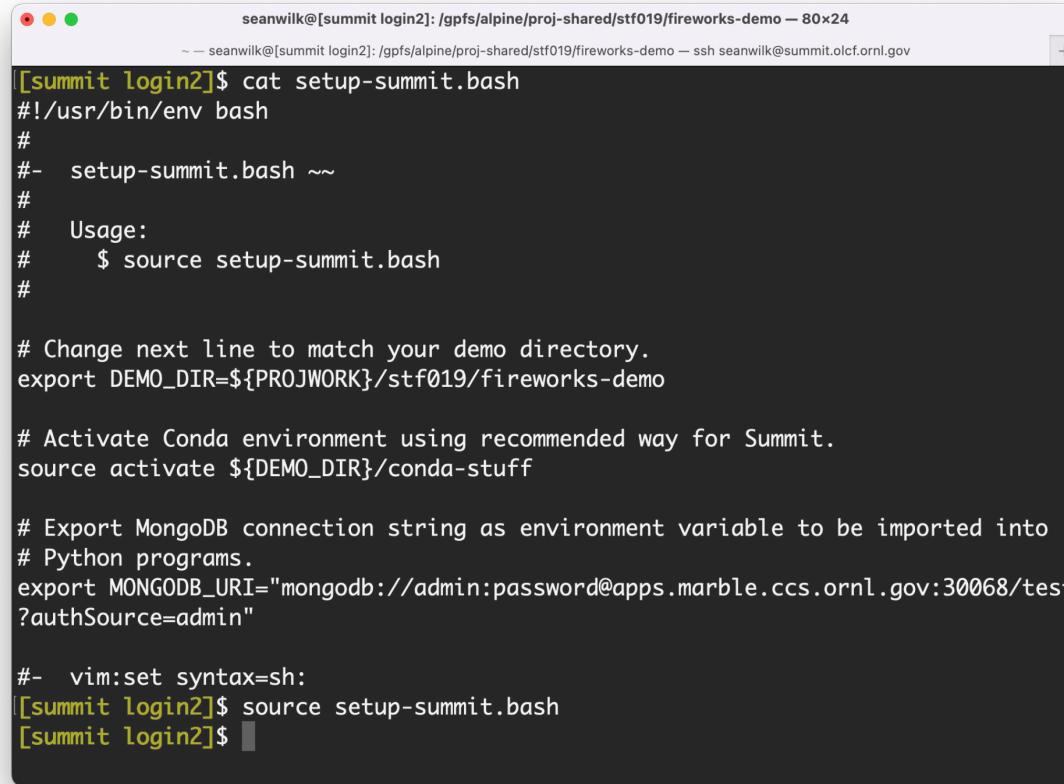
```
((fireworks)stephey@perlmutter:login10:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC> lpad get_fws
[
  {
    "fw_id": 1,
    "created_on": "2023-04-12T05:25:41.651417",
    "updated_on": "2023-04-12T05:26:12.983348",
    "state": "COMPLETED",
    "name": "Unnamed FW"
  },
  {
    "fw_id": 2,
    "created_on": "2023-04-12T05:25:41.651528",
    "updated_on": "2023-04-12T05:26:18.585862",
    "state": "COMPLETED",
    "name": "Unnamed FW"
  },
  {
    "fw_id": 3,
    "created_on": "2023-04-12T05:25:41.651607",
    "updated_on": "2023-04-12T05:27:18.937932",
    "state": "COMPLETED",
    "name": "Unnamed FW"
  }
]
```

# Step 7- Check output

```
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-12-05-36-28-319808> g  
rep -r "attribute" .  
.launcher_2023-04-12-05-37-38-633692/fw_job-7226322.out:pearson correlation coefficients for each attribute  
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-12-05-36-28-319808> c  
at ./launcher_2023-04-12-05-37-38-633692/fw_job-7226322.out  
2023-04-11 22:37:45,003 INFO Hostname/IP lookup (this will take a few seconds)  
2023-04-11 22:37:45,010 INFO Launching Rocket  
2023-04-11 22:37:45,086 INFO RUNNING fw_id: 3 in directory: /pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC  
2023-04-11 22:37:45,096 INFO Task started: ScriptTask.  
pearson correlation coefficients for each attribute  
          0  
age           0.187889  
sex           0.043062  
body_mass_index 0.586450  
blood_pressure   0.441482  
total_cholesterol 0.212022  
ldl_cholesterol    0.174054  
hdl_cholesterol    -0.394789  
total/hdl_cholesterol 0.430453  
log_of_serum_triglycerides 0.565883  
blood_sugar_level 0.382483  
2023-04-11 22:37:46,032 INFO Task completed: ScriptTask  
2023-04-11 22:37:46,048 INFO Rocket finished  
(fireworks)stephey@perlmutter:login26:/pscratch/sd/s/stephey/DOE-HPC-workflow-training/FireWorks/NERSC/block_2023-04-12-05-36-28-319808>
```

# Demo at OLCF with the Python API

# Step 1: \$ source setup-summit.bash



A screenshot of a terminal window titled "seanwilk@[summit login2]: /gpfs/alpine/proj-shared/stf019/fireworks-demo — 80x24". The window shows the script "setup-summit.bash" being viewed with the command "cat setup-summit.bash". The script contains comments and code to set up a Conda environment and MongoDB connection. It ends with a vim configuration line and a final command to source the script.

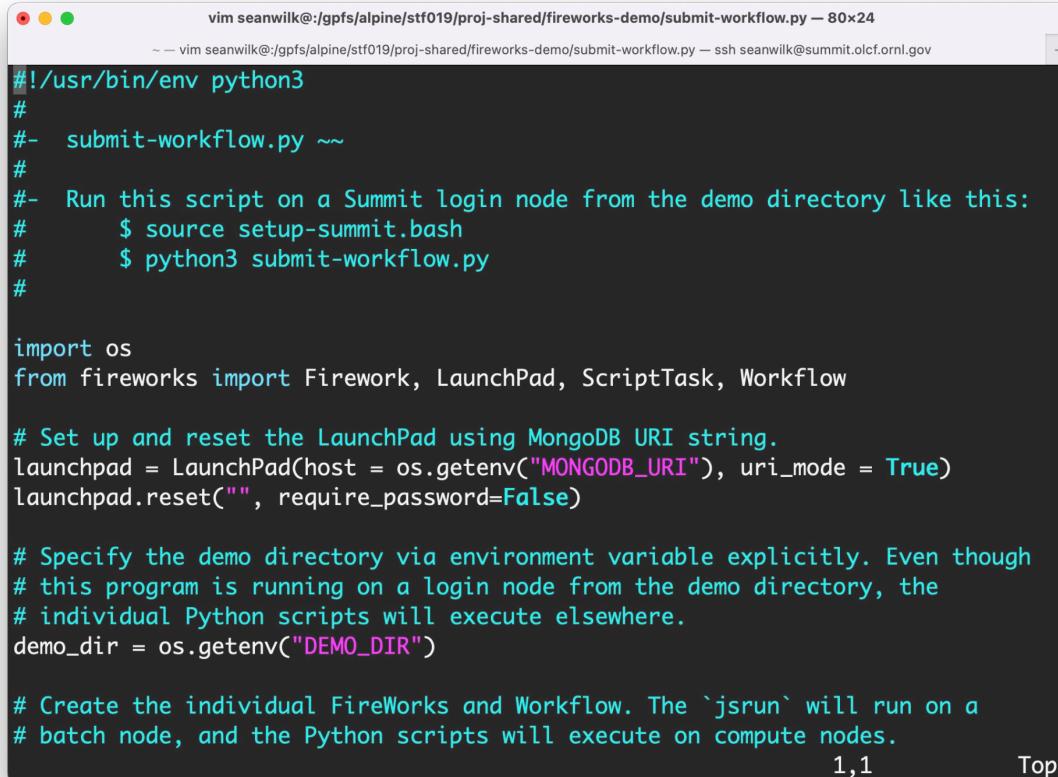
```
[summit login2]$ cat setup-summit.bash
#!/usr/bin/env bash
#
#- setup-summit.bash ~
#
# Usage:
#   $ source setup-summit.bash
#
# Change next line to match your demo directory.
export DEMO_DIR=${PROJWORK}/stf019/fireworks-demo

# Activate Conda environment using recommended way for Summit.
source activate ${DEMO_DIR}/conda-stuff

# Export MongoDB connection string as environment variable to be imported into
# Python programs.
export MONGODB_URI="mongodb://admin:password@apps.marble.ccs.ornl.gov:30068/test
?authSource=admin"

#- vim:set syntax=sh:
[summit login2]$ source setup-summit.bash
[summit login2]$
```

# Step 2: Submit the workflow using Python API



The screenshot shows a terminal window titled "vim seanwilk@:/gpfs/alpine/stf019/proj-shared/fireworks-demo/submit-workflow.py - 80x24". The terminal is running on a Summit login node via SSH. The code in the terminal is a Python script named "submit-workflow.py". The script imports the Firework, LaunchPad, ScriptTask, and Workflow classes from the fireworks module. It sets up a LaunchPad using a MongoDB URI and resets it. It specifies the demo directory via an environment variable. It creates individual FireWorks and Workflows, noting that the jsrun command will run on a batch node while Python scripts execute on compute nodes.

```
#!/usr/bin/env python3
#
#- submit-workflow.py ~-
#
#- Run this script on a Summit login node from the demo directory like this:
#    $ source setup-summit.bash
#    $ python3 submit-workflow.py
#
import os
from fireworks import Firework, LaunchPad, ScriptTask, Workflow

# Set up and reset the LaunchPad using MongoDB URI string.
launchpad = LaunchPad(host = os.getenv("MONGODB_URI"), uri_mode = True)
launchpad.reset("", require_password=False)

# Specify the demo directory via environment variable explicitly. Even though
# this program is running on a login node from the demo directory, the
# individual Python scripts will execute elsewhere.
demo_dir = os.getenv("DEMO_DIR")

# Create the individual FireWorks and Workflow. The `jsrun` will run on a
# batch node, and the Python scripts will execute on compute nodes.
```

1,1

Top

# Step 2: Submit the workflow (continued)

```
vim seanwilk@:/gpfs/alpine/stf019/proj-shared/fireworks-demo/submit-workflow.py -- 80x24
~ vim seanwilk@:/gpfs/alpine/stf019/proj-shared/fireworks-demo/submit-workflow.py - ssh seanwilk@summit.olcf.ornl.gov

# Specify the demo directory via environment variable explicitly. Even though
# this program is running on a login node from the demo directory, the
# individual Python scripts will execute elsewhere.
demo_dir = os.getenv("DEMO_DIR")

# Create the individual FireWorks and Workflow. The `jsrun` will run on a
# batch node, and the Python scripts will execute on compute nodes.
fw1 = Firework(ScriptTask.from_str("jsrun -n 1 -a 1 -c 1 python3 " +
    os.path.join(demo_dir, "step_1_diabetes_preprocessing.py")),
    name = "Step-1")
fw2 = Firework(ScriptTask.from_str("jsrun -n 10 -a 1 -c 1 python3 " +
    os.path.join(demo_dir, "step_2_diabetes_correlation.py")),
    name = "Step-2")
fw3 = Firework(ScriptTask.from_str("jsrun -n 1 -a 1 -c 1 python3 " +
    os.path.join(demo_dir, "step_3_diabetes_postprocessing.py")),
    name = "Step-3")
wf = Workflow([fw1, fw2, fw3], {fw1: fw2, fw2: fw3}, name = "Fireworks demo")

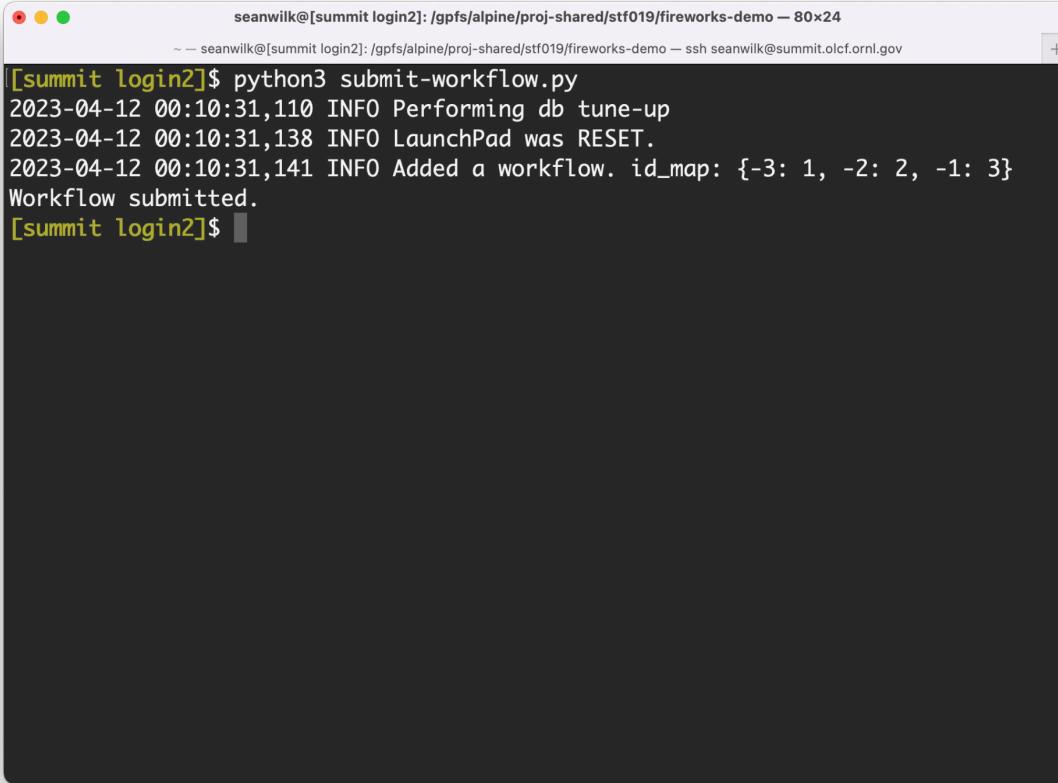
# Store workflow
launchpad.add_wf(wf)
print("Workflow submitted.")

#- vim:set syntax=python:
```

39,1

Bot

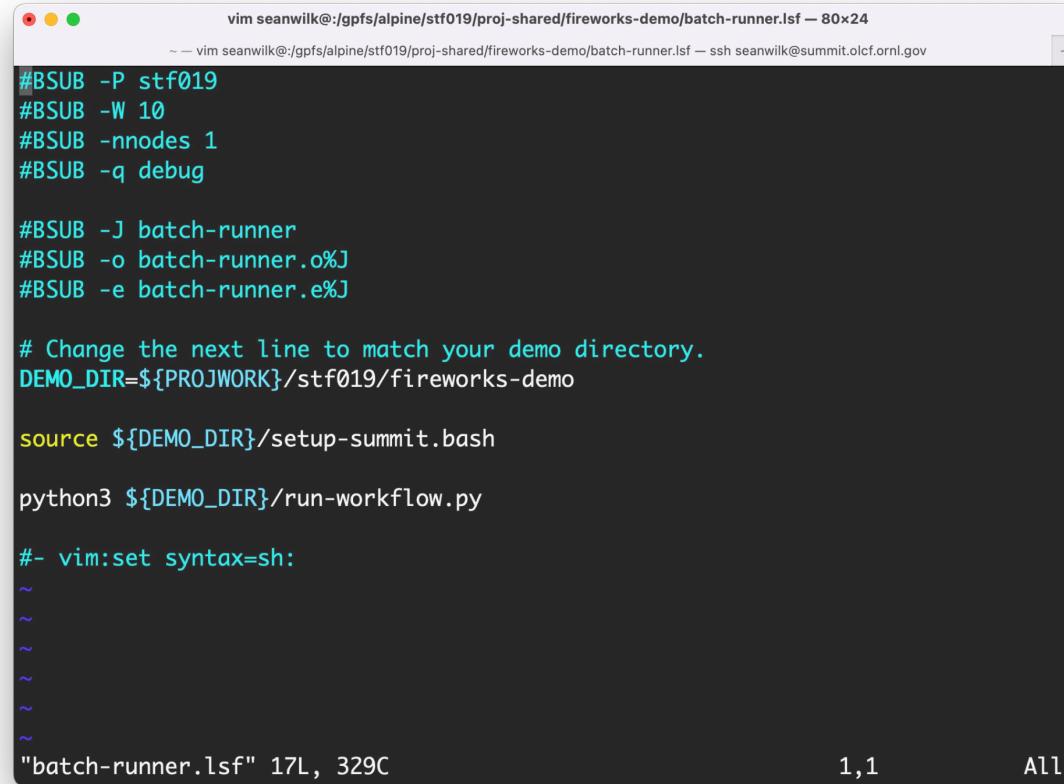
# Step 2: \$ python3 submit-workflow.py



A terminal window titled "seanwilk@[summit login2]: /gpfs/alpine/proj-shared/stf019/fireworks-demo — 80x24" with the command "ssh seanwilk@summit.olcf.ornl.gov". The terminal output shows the execution of "submit-workflow.py" and its log messages:

```
[summit login2]$ python3 submit-workflow.py
2023-04-12 00:10:31,110 INFO Performing db tune-up
2023-04-12 00:10:31,138 INFO LaunchPad was RESET.
2023-04-12 00:10:31,141 INFO Added a workflow. id_map: {-3: 1, -2: 2, -1: 3}
Workflow submitted.
[summit login2]$
```

# Step 3: Submit LSF batch script to execute workflow



A screenshot of a terminal window titled "vim seanwilk@:/gpfs/alpine/stf019/proj-shared/fireworks-demo/batch-runner.lsf - 80x24". The window shows a shell script for submitting a job to the LSF system. The script includes directives for job name, wall time, number of nodes, queue, and output/error files. It also specifies a demo directory and source a setup script before running the workflow. The vim status bar at the bottom indicates the file is "batch-runner.lsf" with 17 lines and 329C.

```
#BSUB -P stf019
#BSUB -W 10
#BSUB -nnodes 1
#BSUB -q debug

#BSUB -J batch-runner
#BSUB -o batch-runner.o%J
#BSUB -e batch-runner.e%J

# Change the next line to match your demo directory.
DEMO_DIR=${PROJWORK}/stf019/fireworks-demo

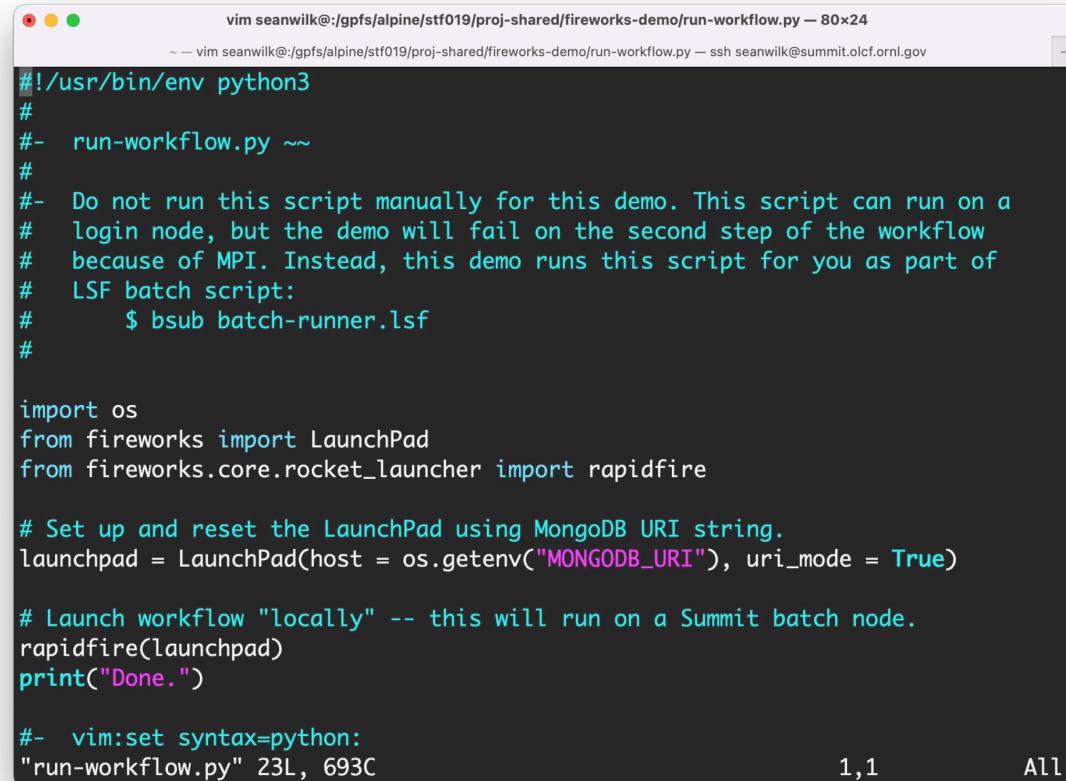
source ${DEMO_DIR}/setup-summit.bash

python3 ${DEMO_DIR}/run-workflow.py

#- vim:set syntax=sh:
```

"batch-runner.lsf" 17L, 329C 1,1 All

# Step 3 (automatic): Batch script launches Python script



A screenshot of a terminal window titled "vim seanwilk@:/gpfs/alpine/stf019/proj-shared/fireworks-demo/run-workflow.py - 80x24". The terminal shows a Python script named "run-workflow.py". The script imports os, LaunchPad, and rapidfire from the fireworks library. It sets up a LaunchPad using a MongoDB URI and launches a workflow locally using rapidfire. A "Done." message is printed to the console. The Vim status bar at the bottom indicates "vim: syntax=python" and file statistics: "run-workflow.py" 23L, 693C.

```
#!/usr/bin/env python3
#
#- run-workflow.py ~
#
#- Do not run this script manually for this demo. This script can run on a
# login node, but the demo will fail on the second step of the workflow
# because of MPI. Instead, this demo runs this script for you as part of
# LSF batch script:
#     $ bsub batch-runner.lsf
#
#
import os
from fireworks import LaunchPad
from fireworks.core.rocket_launcher import rapidfire

# Set up and reset the LaunchPad using MongoDB URI string.
launchpad = LaunchPad(host = os.getenv("MONGODB_URI"), uri_mode = True)

# Launch workflow "locally" -- this will run on a Summit batch node.
rapidfire(launchpad)
print("Done.")

#- vim:set syntax=python:
```

1,1

All

# Step 3: \$ bsub batch-runner.lsf

```
seanwilk@[summit login2]: /gpfs/alpine/proj-shared/stf019/fireworks-demo — 80x24
~ — seanwilk@[summit login2]: /gpfs/alpine/proj-shared/stf019/fireworks-demo — ssh seanwilk@summit.olcf.ornl.gov
[summit login2]$ bsub batch-runner.lsf
Job <2863155> is submitted to queue <debug>.
[summit login2]$ bjobs
JOBID    USER      STAT  SLOTS  QUEUE      START_TIME      FINISH_TIME  JOB_N
AME
2863155  seanwilk  PEND      -   debug          -              -              -   batch
-runner
[summit login2]$ bjobs
JOBID    USER      STAT  SLOTS  QUEUE      START_TIME      FINISH_TIME  JOB_N
AME
2863155  seanwilk  RUN       43   debug  Apr 12 00:20  Apr 12 00:30  batch
-runner
[summit login2]$ bjobs
No unfinished job found
[summit login2]$ ls
all_coeffs.npy           run-workflow.py
batch-runner.e2863155     setup-summit.bash
batch-runner.lsf          step_1_diabetes_preprocessing.py
batch-runner.o2863155     step_2_diabetes_correlation.py
conda-stuff               step_3_diabetes_postprocessing.py
launcher_2023-04-12-04-21-13-330351 submit-workflow.py
launcher_2023-04-12-04-21-16-124717 x_diabetes.npy
launcher_2023-04-12-04-21-17-193559 y_diabetes.npy
[summit login2]$
```

# Check output file to verify results

```
vim seanwilk@:/gpfs/alpine/stf019/proj-shared/fireworks-demo/batch-runner.o2863155 — 80x24
~ — vim seanwilk@:/gpfs/alpine/stf019/proj-shared/fireworks-demo/batch-runner.o2863155 — ssh seanwilk@summit.olcf.ornl.gov
2023-04-12 00:21:17,216 INFO RUNNING fw_id: 1 in directory: /gpfs/alpine/stf019/
proj-shared/fireworks-demo/launcher_2023-04-12-04-21-17-193559
2023-04-12 00:21:17,220 INFO Task started: ScriptTask.
[ 0.188  0.043  0.586  0.441  0.212  0.174 -0.395  0.43   0.566  0.382]
Pearson correlation coefficients for each attribute
          0
age        0.187889
sex        0.043062
body_mass_index  0.586450
blood_pressure    0.441482
total_cholesterol  0.212022
ldl_cholesterol    0.174054
hdl_cholesterol    -0.394789
total/hdl_cholesterol  0.430453
log_of_serum_triglycerides  0.565883
blood_sugar_level   0.382483
2023-04-12 00:21:18,336 INFO Task completed: ScriptTask
2023-04-12 00:21:18,349 INFO Rocket finished
Done.

-----
Sender: LSF System <lsfadmin@batch2>
Subject: Job 2863155: <batch-runner> in cluster <summit> Done
51,1           73%
```

# Answer to our diabetes question

- In this dataset, body mass index (BMI) and triglycerides are most highly correlated with diabetes risk
- Interestingly HDL cholesterol is anti-correlated with diabetes (this is because it is “good” cholesterol!)

```
pearson correlation coefficients for each attribute
    0
age           0.187889
sex           0.043062
body_mass_index 0.586450
blood_pressure 0.441482
total_cholesterol 0.212022
ldl_cholesterol 0.174054
hdl_cholesterol -0.394789
total/hdl_cholesterol 0.430453
log_of_serum_triglycerides 0.565883
blood_sugar_level 0.382483
```

# FireWorks wrap up

- FireWorks is a powerful workflow tool that can be run at ALCF, NERSC, and OLCF
- It requires a MongoDB – all centers can provide guidance for setting one up
- FireWorks is best suited for high-throughput workflows, but can be used to describe multi-resource workflows like the one in our demo
- FireWorks can support multi-node MPI tasks
- FireWorks can either be used with .yaml files (language-agnostic) or with a Python API
- Check out the [FireWorks documentation](#) for more in-depth info
- If you have questions, join the [FireWorks user community forum](#)

# Extra slides

# Configuring your LaunchPad

- Once you have a MongoDB, the next step is to configure your FireWorks LaunchPad to use your MongoDB
- You will be prompted to enter the following information:
  - Hostname/IP of your MongoDB
  - MongoDB port number
  - Your database name
  - Your admin username
  - Your admin password
  - Other fields are optional
- If successfully, this will write a `my_launchpad.yaml` config file
- You can test that your launchpad is working with a command like  
`lpad reset` or `lpad -h` to show the full help menu

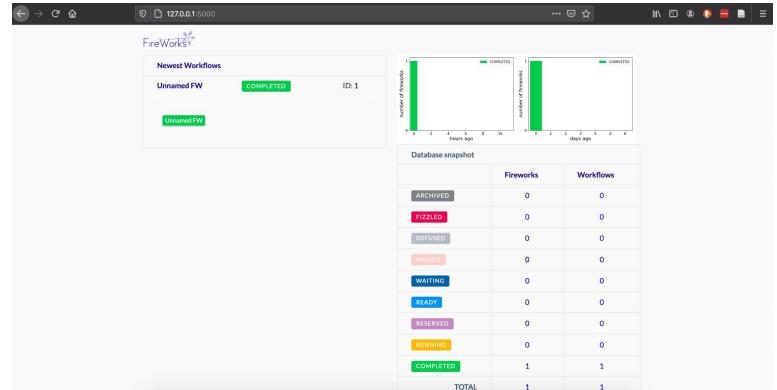
## qlaunch vs. rlaunch

- FireWorks can run tasks in either a HPC queue with qlaunch or locally on the FireServer node with rlaunch
- This can get confusing when using a queue adapter with the `rocket_launch: rlaunch key`
- Why both qlaunch and rlaunch?
- qlaunch launches the job, and once running, rlaunch launches the workflow in either `rapidfire` or `singleshot` mode

# singleshot vs. rapidfire

- FireWork workflows can be configured to run in either singleshot or rapidfire mode
- This can be get confusing when submitting a job via qlaunch
- In some cases, you may want to submit a qlaunch singleshot job but within the my\_qdapter.yaml, the rocket\_launch key may contain rlaunch rapidfire -> this is a job you have submitted once, but within the job, the task may run many times
- Conversely, you may want to submit a qlaunch rapidfire job but within the my\_qapdater.yaml, the rocket\_launch key main contain rlaunch singleshot -> this is a job that will be submitted many times, but within the job, the task will only run once

# Using the FireWorks dashboard



We didn't have time to cover it in our tutorial, but FireWorks has a handy dashboard

To use it, you can follow the instructions here:

<https://docs.nersc.gov/jobs/workflow/fireworks/#display-the-fireworks-dashboard>

Note these instructions are for NERSC Cori but this same port-forwarding procedure can be used on any system

# Simple mode vs. reservation mode

In this demo we chose to showcase the [simple queue launch mode](#)

FireWorks also has a more [advanced reservation mode](#)

Situation	Simple Queue Launching	Reservation Queue Launching
write/submit queue script	write generic script using QueueAdapter file alone	1. reserve a FW from the database 2. use FW's spec to modify queue script
queue manager runs queue script	determine a FW to run and run it	run the reserved FW
job is deleted from queue	no action needed by the user	any affected reserved jobs must be unreserved by user using detect_unreserved
run multiple FWs in one script	supported	currently unsupported
offline mode	unsupported	supported