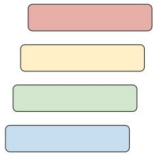
Wrangling Complex Notebook Workflows with Floability

Douglas Thain, on behalf of the Floability team:

Saiful Islam, Talha Aziz, Shahadat Hossain, Raza Ahmad, Furquan Baig, Tanu Malik, Kevin Lannon, Shaowen Wang

Throughput Computing 2025
Madison, WI July 2024

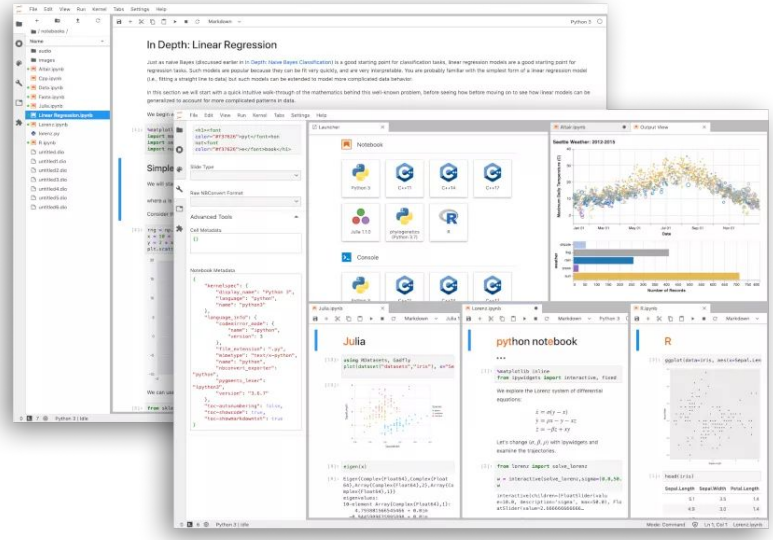


floability

Two Different Worlds of Computing?

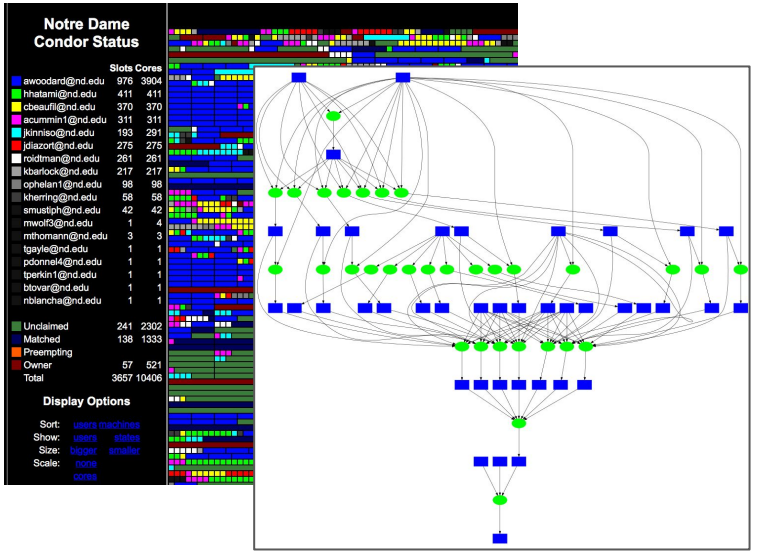
Interactive Notebooks

Graphical, Interactive, Personal, **Limited**



High Throughput Workflows

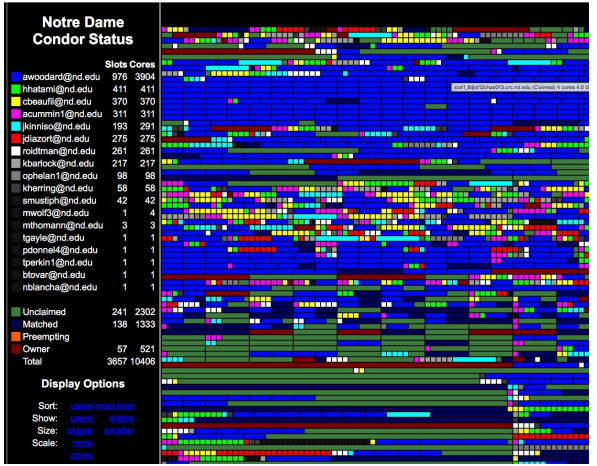
Scripting, Batch, Shared, **Scalable**



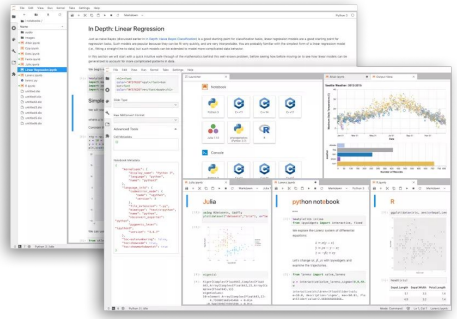
Nobody* Starts with High Throughput!

They begin by writing Python in a notebook on the laptop.
And then they share and publish that notebook with others.
After a while… they bump up against the limits of one node.

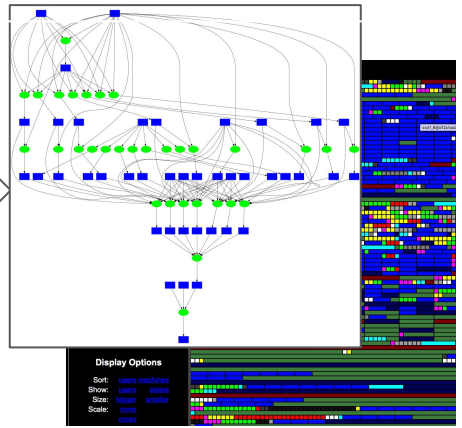
???



The **Floability Project** aims to enable the rapid and portable deployment of notebooks expressing complex scientific workflows across a wide range of cyberinfrastructure.



floability



CSSI Frameworks: From Notebook to Workflow and Back Again

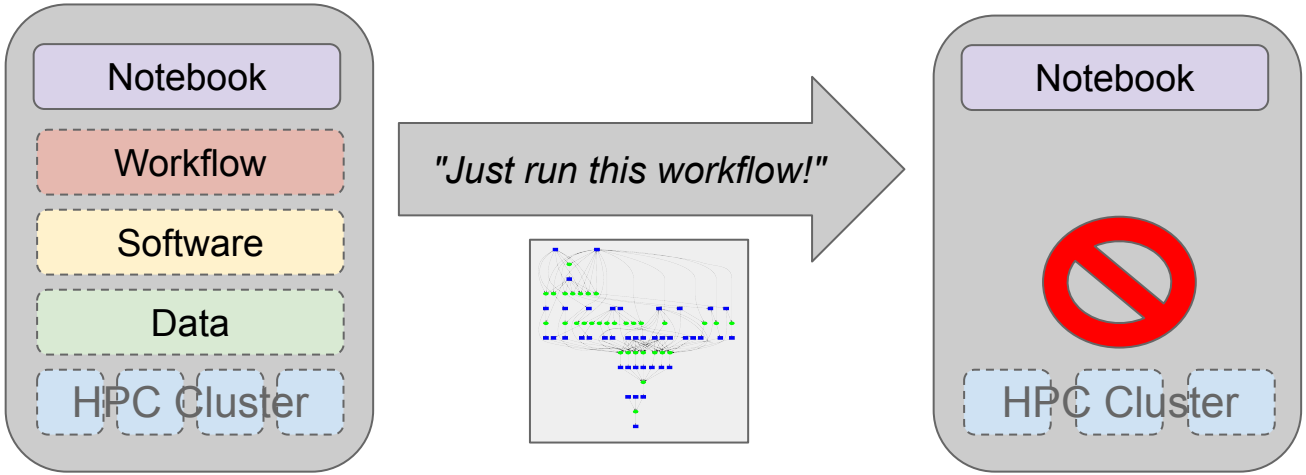
NSF

UNIVERSITY OF NOTRE DAME

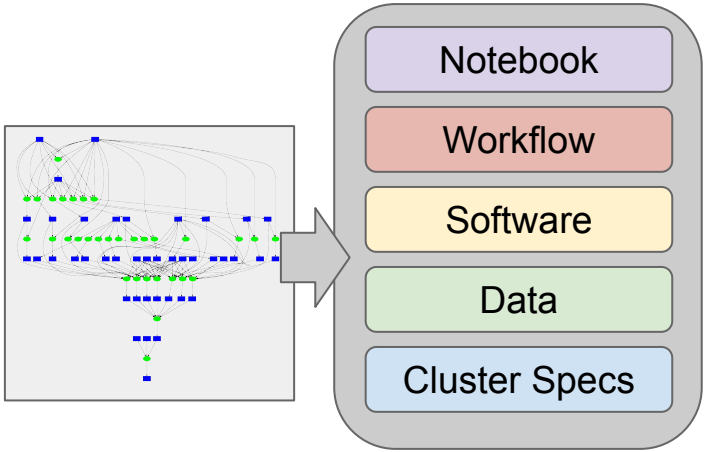
Mizzou
University of Missouri

ILLINOIS

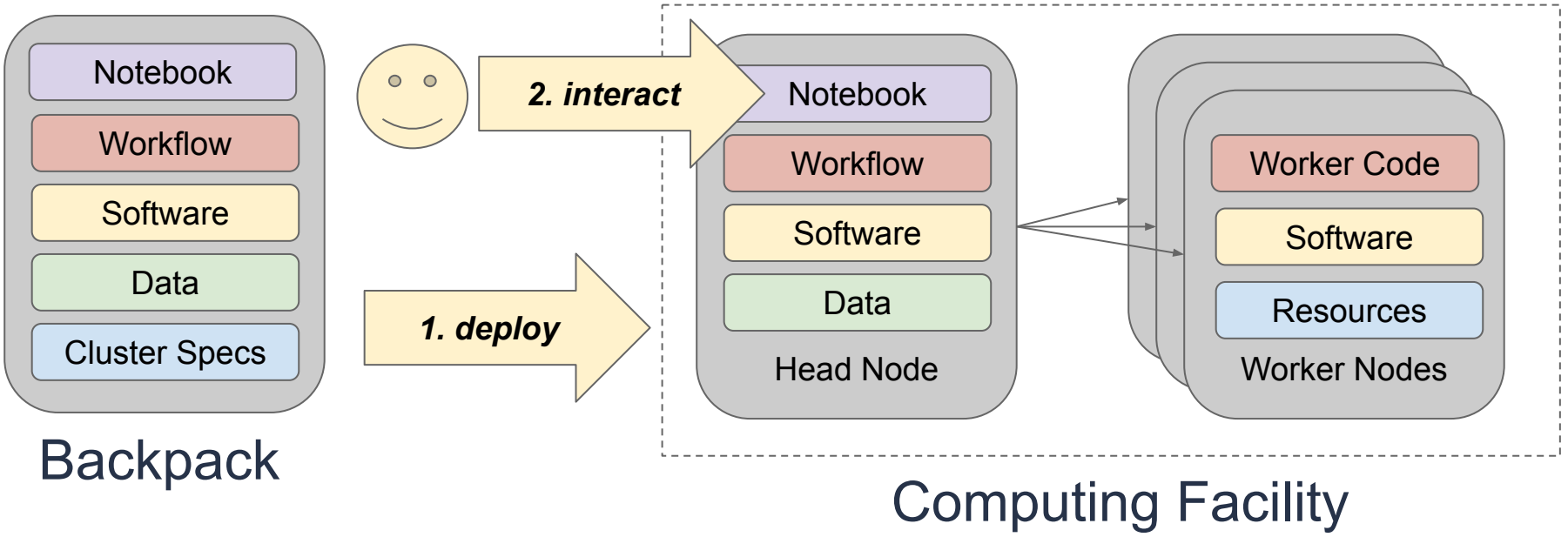
You cannot deploy a notebook workflow effectively without all the supporting environment.



A **backpack** contains everything needed to deploy a notebook workflow at large scale.

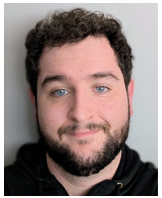


Floability deploys a **backpack** into a facility.



Example: DV5 CMS Analysis Application

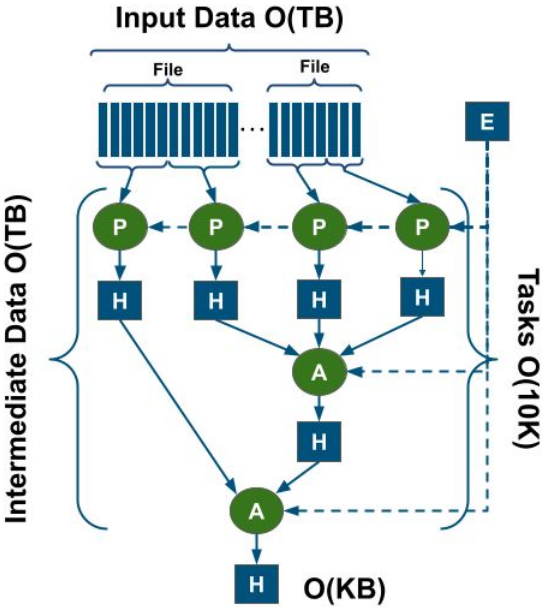
http://dx.doi.org/10.1109/SC41406.2024.00068

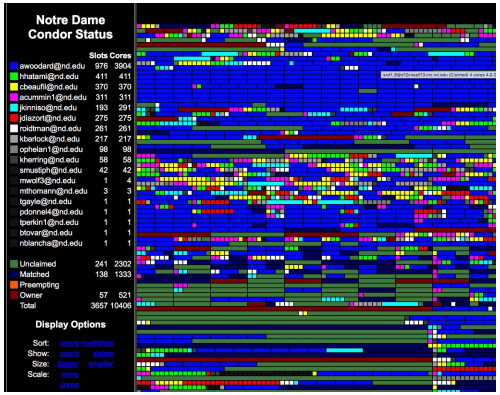


Kevin Lannon **Connor Moore**

```
In [3]: def analysis(events):
            warnings.filterwarnings("ignore", module="coffea.*")

            dataset = events.metadata["dataset"]
            events["PFCands", "pt"] = events.PFCands.pt * events.PFCands.puppiW

            cut_to_fix_softdrop = ak.num(events.FatJet.constituents.pf, axis=2)
            events = events[ak.all(cut_to_fix_softdrop, axis=1)]

            trigger = ak.zeros_like(ak.firsts(events.FatJet.pt), dtype="bool")
            for t in triggers["2017"]:
                if t in events.HLT.fields:
                    trigger = trigger | events.HLT[t]
            trigger = ak.fill_none(trigger, False)

            events["FatJet", "num_fatjets"] = ak.num(events.FatJet)

            goodmuon = (
                (events.Muon.pt > 10)
                & (abs(events.Muon.eta) < 2.4)
                & (events.Muon.pfRelIso04_all < 0.25)
                & events.Muon.looseId
            )

            nmuons = ak.sum(goodmuon, axis=1)

In [5]: m = DaskVine(
            [9123, 9128],
            name=f"{os.environ.get("VINE_MANAGER_NAME")}",
        )

In [6]: computed = dask.compute(
            tasks,
            scheduler=m,
            ss_disable=True,
            ces={"cores": 
            ces_mode=None,

        pute done.")
```

File display





Consumes 1.5TB Data
Submits 17K Tasks
Uses 2400 cores, 200 nodes.

jupyter

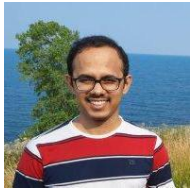
dask

TaskVine

HTCondor

Example: Surface Ocean Heat (CESM2)

https://github.com/floability/floability-examples/tree/main/cesm_oceanheat



Harsha Hampapura

Setting Dask Schedulers to Use DaskVine

```
import ndcctools.taskvine as vine
from functools import partial
import os

m = vine.DaskVine(
    [9123, 9500],
    name=f"{os.environ.get("VINE_MANAGER_NAME")}",
)

vine_scheduler = partial(m.get, progress_disable = True)

dask.config.set(scheduler=vine_scheduler)
```

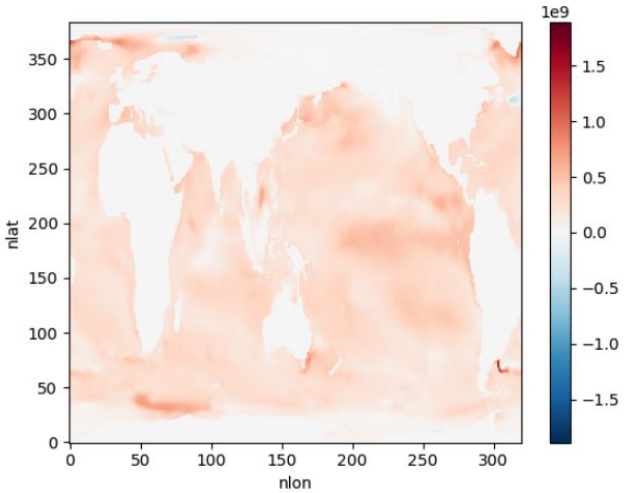
Data: CESM2 LENS 1850-2100
Tasks: 2800+ parallel jobs
Tools: Xarray + Dask/TaskVine
Notebook Size: 1.3 MB
Backpack Size: 1.5 MB

Has the surface ocean heat content increased with time for January ?

```
hist_ocean_avgheat_ano = hist_ocean_avgheat.isel(time=1) - hist_ocean_avgheat.i
```

```
%%time
hist_ocean_avgheat_ano.plot()
```

```
CPU times: user 4.77 s, sys: 932 ms, total: 5.7 s
Wall time: 53.6 s
<matplotlib.collections.QuadMesh at 0x7f78f3f69f70>
```



NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

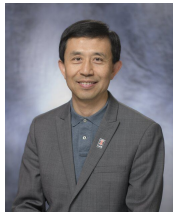
OSDF
CHTC

Example: Aging Dams Simulation Workflow

https://github.com/floability/floability-cli/tree/distributed-iguide-gis-aging-dams/example/iguide-gis-ag

I-GUIDE
Institute for Geospatial Understanding through an Integrative Discovery Environment

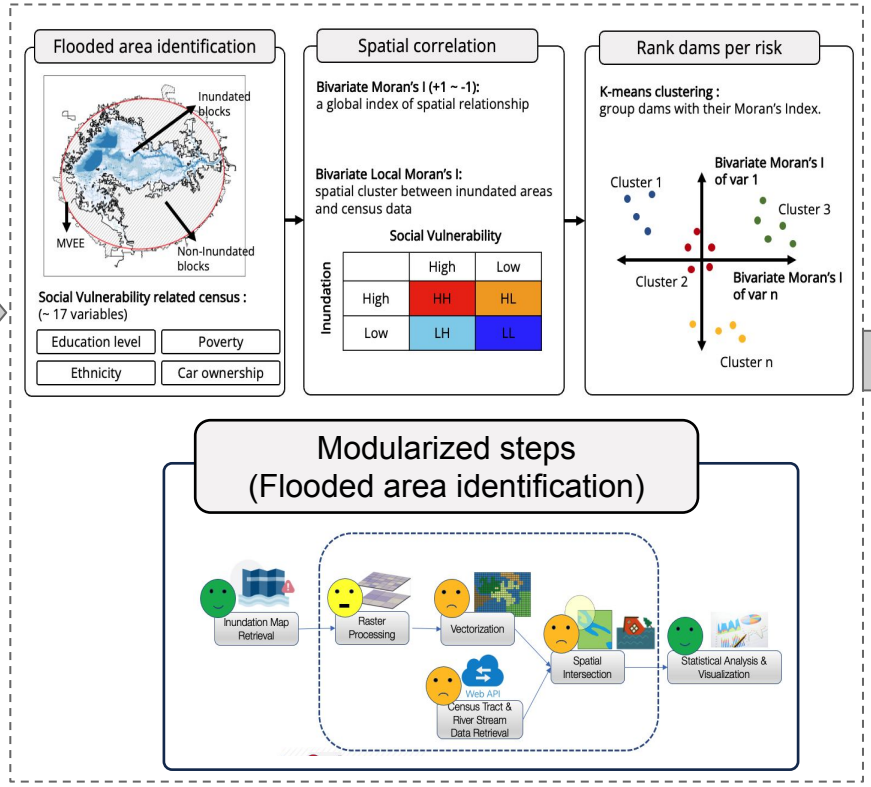
Furquan Baig

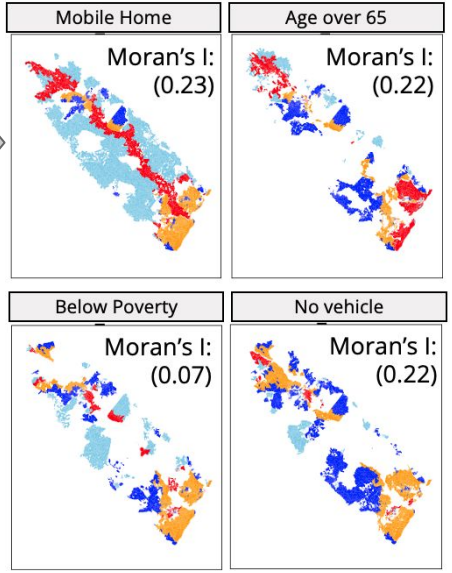


Shaowen Wang

Analyze Downstream Impacts of Dam Failures to Critical Infrastructure, Vulnerable Populations & Ecosystems

Backpack captured but not distributed (yet)





So What's Actually in a Backpack? (V1)

https://github.com/floability/floability-examples/tree/main/cms-physics-dv5

```
➔ workflow
   ◆ cms-physics-dv5.ipynb
➔ software
   ◆ environment.yml
   ◆ worker-environment.yml
➔ data
   ◆ data.yml
   ◆ samples/qcd/.../nano_mc2017_11.root
   ◆ samples/diboson/.../nanomc2017_6.root
➔ compute
   ◆ compute.yml
```

So What's Actually in a Backpack? (V1)

https://github.com/floability/floability-examples/tree/main/cms-physics-dv5

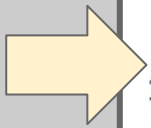
- **workflow**
  - **cms-physics-dv5.ipynb**
- software
  - environment.yml
  - worker-environment.yml
- data
  - data.yml
  - samples/qcd/.../nano_mc2017_11
  - samples/diboson/.../nanomc2017
- compute
  - compute.yml

```
def analysis(events):
    warnings.filterwarnings("ignore", module="coffea.*"

    dataset = events.metadata["dataset"]
    events["PFCands", "pt"] = events.PFCands.pt * event

    cut_to_fix_softdrop = ak.num(events.FatJet.constitu
    events = events[ak.all(cut_to_fix_softdrop, axis=1)

    trigger = ak.zeros_like(ak.firsts(events.FatJet.pt)
    for t in triggers["2017"]:
        if t in events.HLT.fields:
            trigger = trigger | events.HLT[t]
    trigger = ak.fill_none(trigger, False)

    events["FatJet", "num_fatjets"] = ak.num(events.Fat

    goodmuon = (
        (events.Muon.pt > 10)
        & (abs(events.Muon.eta) < 2.4)
        & (events.Muon.pfRelIso04_all < 0.25)
        & events.Muon.looseId
    )

    nmuons = ak.sum(goodmuon, axis=1)
```

So What's Actually in a Backpack?

https://github.com/floability/floability-examples/tree/main/cms-physics-dv5

- workflow
  - cms-physics-dv5.ipynb
- software
  - **environment.yml**
  - worker-environment.yml
- data
  - data.yml
  - samples/qcd/.../nano_mc2017_11.
  - samples/diboson/.../nanomc2017_
- compute
  - compute.yml

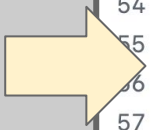


```yaml
name: my_mdv5_env

channels:
  - conda-forge

dependencies:
  - python=3.12
  - cloudpickle
  - ndcctools
  - coffea<2024.4.2
  - dask<2024.5.2
  - dask-awkward<2024.5.0
  - dask-core<2024.5.0
  - dask-histogram<2024.5.0
  - fsspec
  - pip:
    - fastjet<3.4.2.2
```

So What's Actually in a Backpack? (V1)

https://github.com/floability/floability-examples/tree/main/cms-physics-dv5

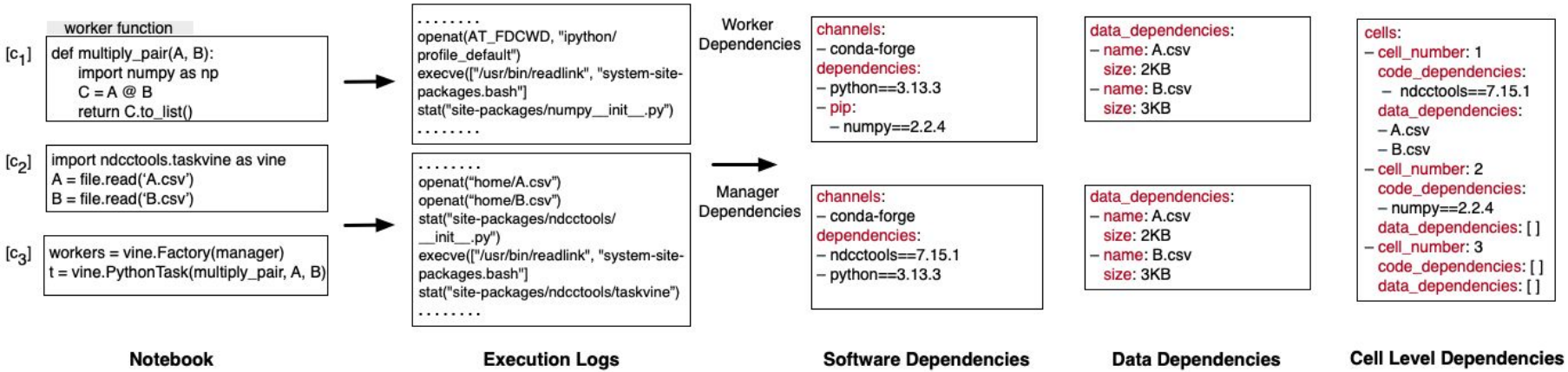
- workflow
  - cms-physics-dv5.ipynb
- software
  - environment.yml
  - **worker-environment.yml**
- data
  - data.yml
  - samples/qcd/.../nano_mc2017_11.root
  - samples/diboson/.../nanomc2017_6.rc
- compute
  - compute.yml



```
41    - conda-forge::pytz==2025.2
42    - conda-forge::rich==14.0.0
43    - conda-forge::scipy==1.15.2
44    - conda-forge::setuptools==78.1.0
45    - conda-forge::tblib==3.1.0
46    - conda-forge::toolz==1.0.0
47    - conda-forge::tqdm==4.67.1
48    - conda-forge::uhi==0.5.0
49    - conda-forge::uproot==5.6.0
50    - conda-forge::wheel==0.45.1
51    - conda-forge::xxhash==0.8.3
52    - conda-forge::yaml==0.2.5
53    - conda-forge::zstandard==0.23.0
54    - pip==25.0.1
55    - python==3.12.10
56    - pip:
57      - debugpy==1.8.14
58      - fastjet==3.4.2.1
59      - ipython-pygments-lexers==1.1.1
60      - ipython==9.1.0
61      - jsonschema==4.23.0
62      - jupyter-client==8.6.3
63      - jupyter-core==5.7.2
64      - matplotlib-inline==0.1.7
65      - nbformat==5.10.4
66      - vector==1.6.1
```

How do you find software/data deps?

Use the **SciUnit** technology to instrument a running notebook to observe the Python dependencies of the manager/worker.



Fils, Gabriel, Zhihao Yuan, and Tanu Malik. "Sciunits: Reusable research objects."
In *2017 IEEE 13th International Conference on e-Science (e-Science)*, pp. 374-383. IEEE, 2017.

So What's Actually in a Backpack? (V1)

https://github.com/floability/floability-examples/tree/main/cms-physics-dv5

- workflow
  - cms-physics-
- software
  - environment.
  - worker-envir
- **data**
  - **data.yml**
  - **samples/qcd/.../nano_mc2017_11.root**
  - **samples/diboson/.../nanomc2017_6.root**
- compute
  - compute.yml

```yaml
data:
  - name: "diboson_zz_6"
    source_type: "backpack"
    source: "data/samples/diboson/zz/nano_mc2017_6.root"
    target_location: "data/samples/diboson/zz/nano_mc2017_6.root"

  - name: "diboson_zz_9"
    source_type: "backpack"
    source: "data/samples/diboson/zz/nano_mc2017_9.root"
    target_location: "data/samples/diboson/zz/nano_mc2017_9.root"
```

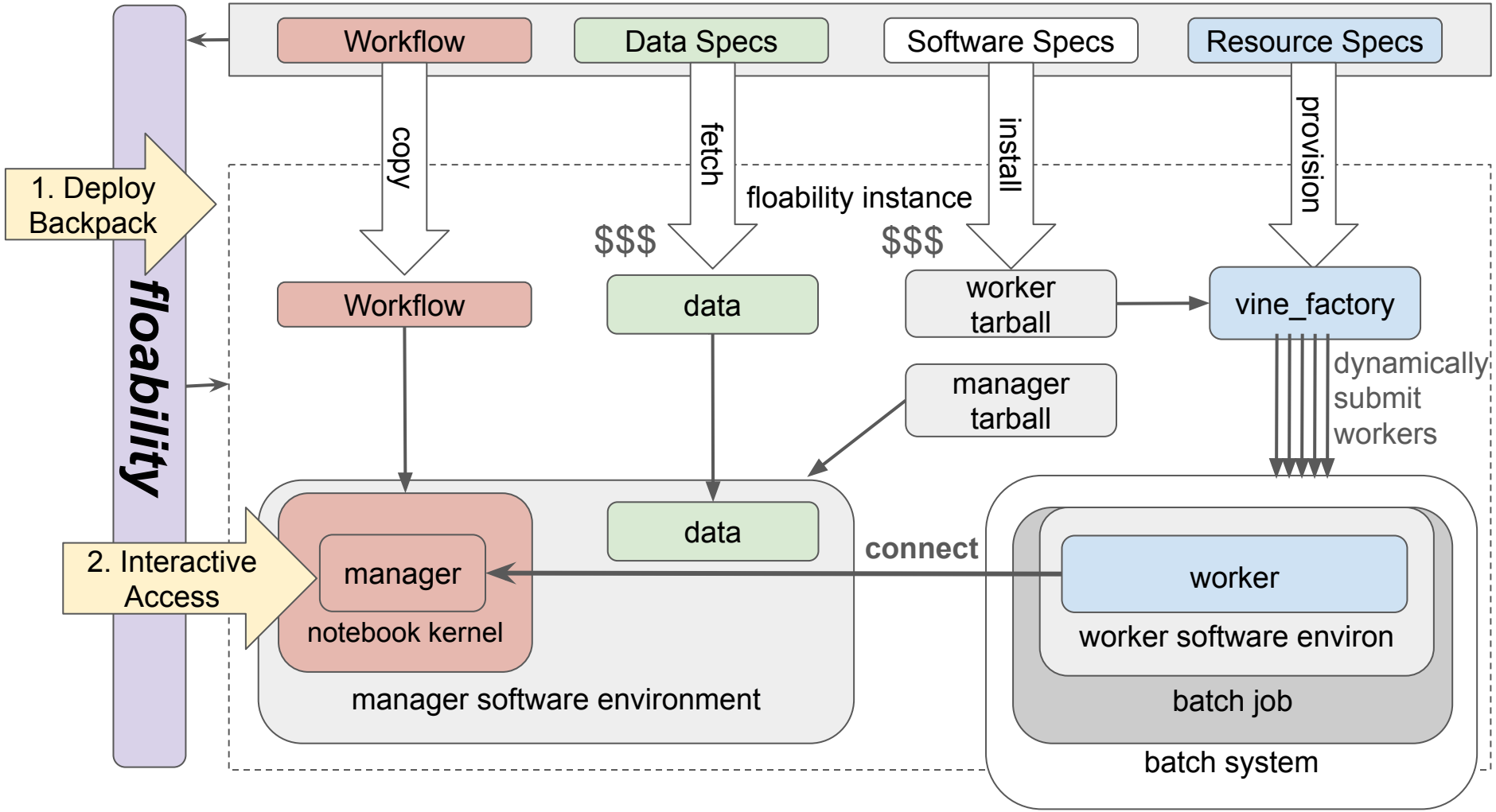
Opportunity to connect Pelican!

So What's Actually in a Backpack? (V1)

https://github.com/floability/floability-examples/tree/main/cms-physics-dv5

- workflow
  - cms-physics-dv5.ipynb
- software
  - environment.yml
  - worker-environment.yml
- data
  - data.yml
  - samples/qcd/.../nano_mc2017_
  - samples/diboson/.../nanomc20
- **compute**
  - **compute.yml**

```
1    vine_factory_config:
2      min-workers: 2
3      max-workers: 400
4      cores: 4
5      memory: 1024
6      disk: 2000
```


Workflow
Data Specs
Software Specs
Resource Specs
copy
fetch
install
provision
floability
1. Deploy Backpack
floability instance
$$$
$$$
Workflow
data
worker tarball
vine_factory
manager tarball
dynamically submit workers
2. Interactive Access
manager
notebook kernel
data
connect
worker
worker software environ
batch job
manager software environment
batch system

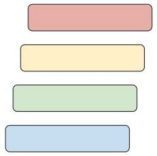
Deployment is a Work in Progress

Workflow	ND CRC	Purdue Anvil	UT Stampede3	AWS Cluster	OSPool
DV5	✔	✔	✔	✔	
DConv	✔	✔	✔	✔	
CTrend	✔	✔	✔	✔	
CESM Ocean Heat	✔	✔	✔	✔	
Montage	✔				

https://github.com/floability/floability-examples

Deployment to Multiple Sites: floability run

Workflow	Metric	ND CRC		Purdue Anvil		UT Stampede3		AWS Cluster	
		First	Repeat	First	Repeat	First	Repeat	First	Repeat
DConv	Total Runtime	292	136	744	255	684	330	216	88
	Env Creation (M + W)	132	0	516	0	307	0	107	0
	Env Extraction	26	29	36	39	116	116	26	26
	Notebook Execution	116	98	174	198	229	171	66	51
CTrend	Total Runtime	265	107	1052	304	405	201	302	216
	Env Creation (M + W)	159	0	763	0	217	0	133	0
	Env Extraction	30	29	44	44	97	90	30	32
	Notebook Execution	57	64	225	240	75	101	127	172
DV5	Total Runtime	370	70	1212	281	827	218	500	89
	Env Creation (M + W)	303	0	973	0	598	0	242	0
	Env Extraction	28	29	43	44	105	83	29	37
	Notebook Execution	25	28	172	217	103	115	215	40



floability

floability.github.io

Overall Project Status:

- Year one of a four year translational project.
- MVP of the Floability tool is published via Conda:
  - `conda install -c conda-forge floability`
- Currently gathering and testing applications at multiple sites.
  - OSPool – Need to be able to connect from worker to manager.
- Looking for applications and (brave) initial users.

Looking Ahead to New Challenges

- Estimate and Encapsulate "Whole Workflow" Requirements
  - Find more ways to say NO prior to execution: available disk space, worker capacity, network utilization, cluster architecture…
- Reconciling Interactive and Batch Allocation
  - The interactive allocation is not useful without the batch allocation, and vice versa. Both have limited capacity to be managed!
- Capture Site Specializations
  - Every facility is a snowflake. Can we capture functional differences (network, storage, policy) between facilities in a constructive way?
- Exploit Concurrency in Notebook Structure
  - Instead of writing in a parallel framework, can we infer independent tasks from the cell structure itself?

floability

floability.github.io

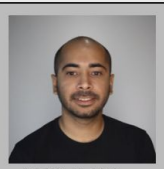
```
conda install -c conda-forge floability
```


Prof. Douglas Thain
Principal Investigator
University of Notre Dame

Prof. Tanu Malik
co-PI
University of Missouri

Prof. Kevin Lannon
co-PI
University of Notre Dame

Prof. Shaowen Wang
co-PI
University of Illinois



Md Saiful Islam
Ph.D. Student
University of Notre Dame

Talha Azaz
MS. Student
DePaul University

Ben Tovar
Research Software Engineer
University of Notre Dame

A S M Shahadat Hossain
Ph.D. Student
University of Missouri

Raza Ahmad
Ph.D. Student
DePaul University

Dr. Furqan Baig
Research Programmer
University of Illinois Urbana Champaign

This work was supported in part by NSF grant 2411436 CSSI Frameworks: From Notebook to Workflow and Back Again