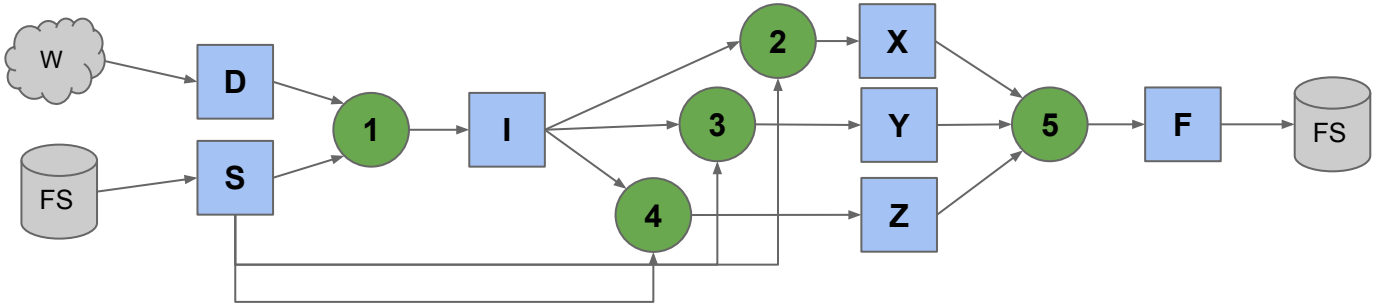
Reshaping High Energy Physics Applications for Near-Interactive Execution Using TaskVine

Barry Sly-Delgado, Ben Tovar

Workflow Transitions

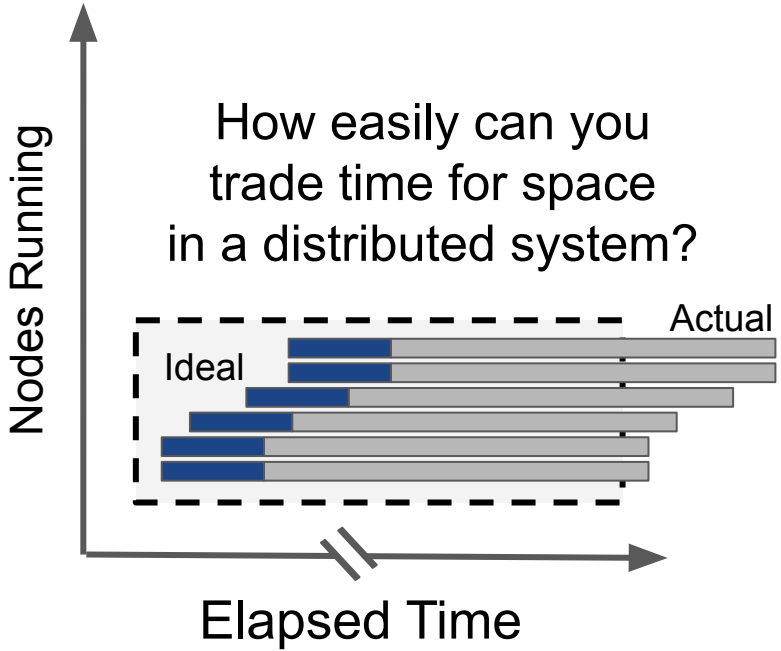
- Changes to hardware and software configurations for a given workflow can greatly impact its runtime performance.
- Such changes allow users to transition longer running high-throughput workflows towards shorter running highly concurrent workloads
- Notably, as the individual tasks of a workflow become shorter, execution tends to be dominated by task startup.
- This startup includes transferring and loading data dependencies.



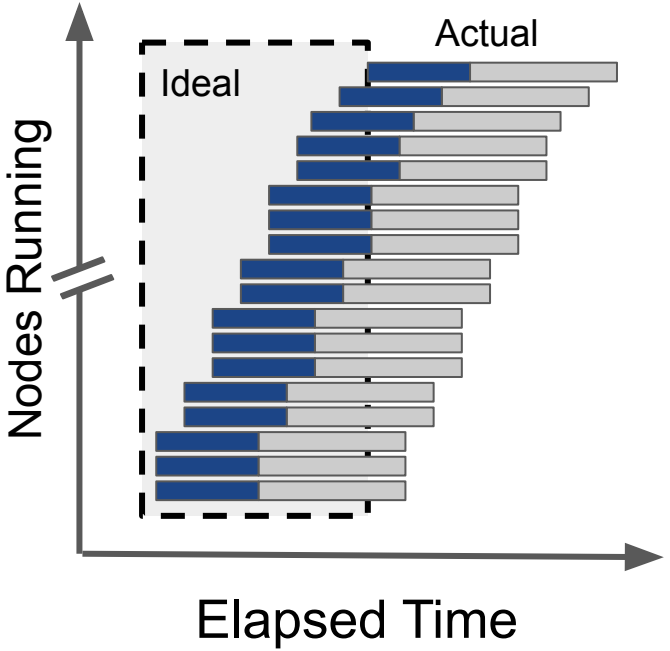
Workflow Reshaping

High Throughput:
N nodes for 60 minutes

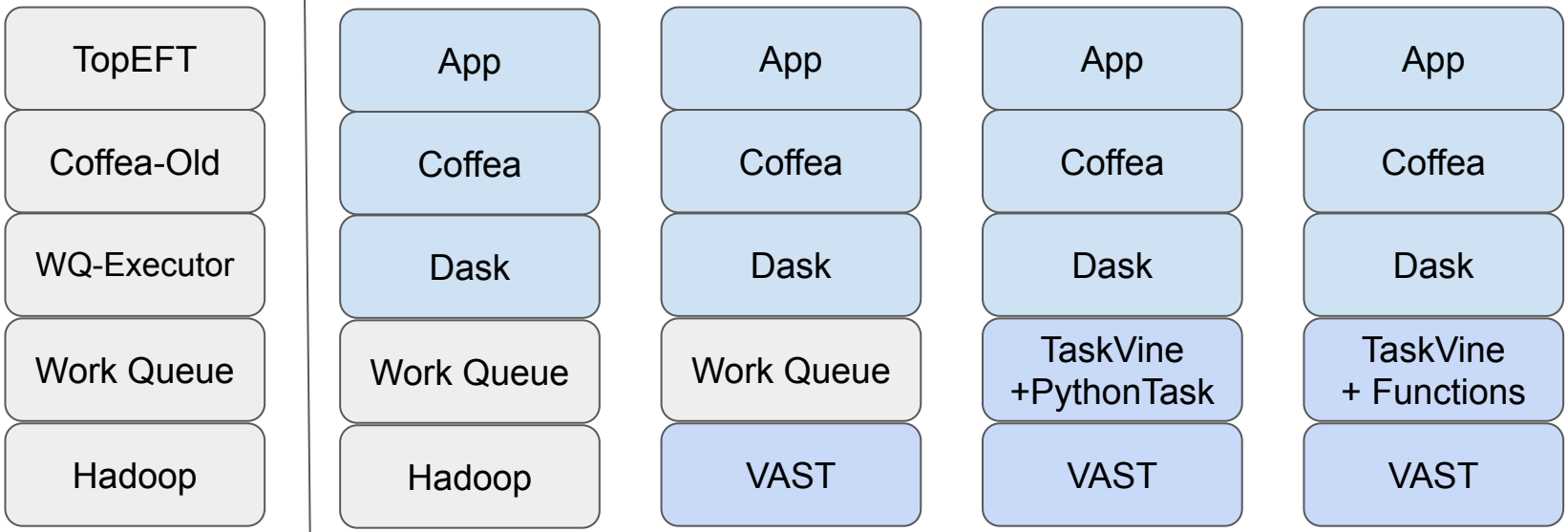
How easily can you
trade time for space
in a distributed system?



High Concurrency
N*10 nodes for 6 minutes



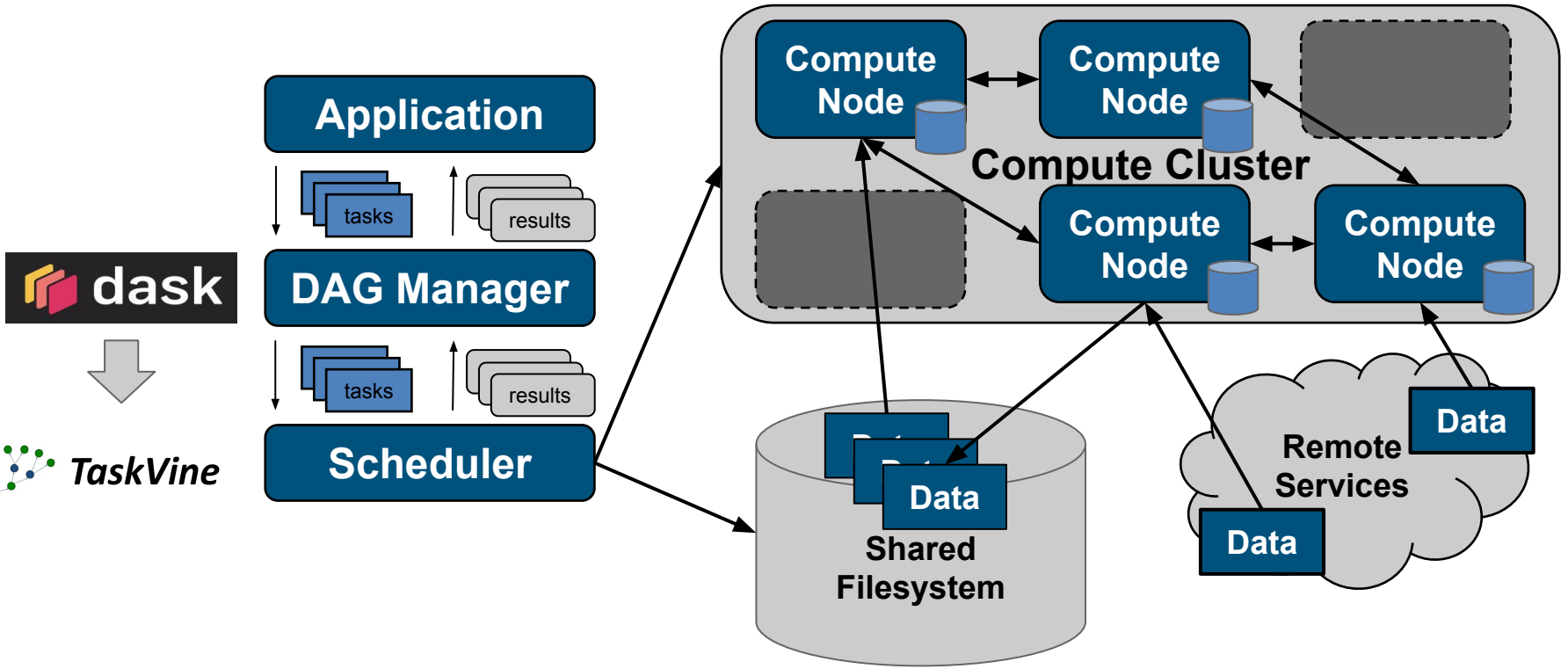
Application Stack Evolution



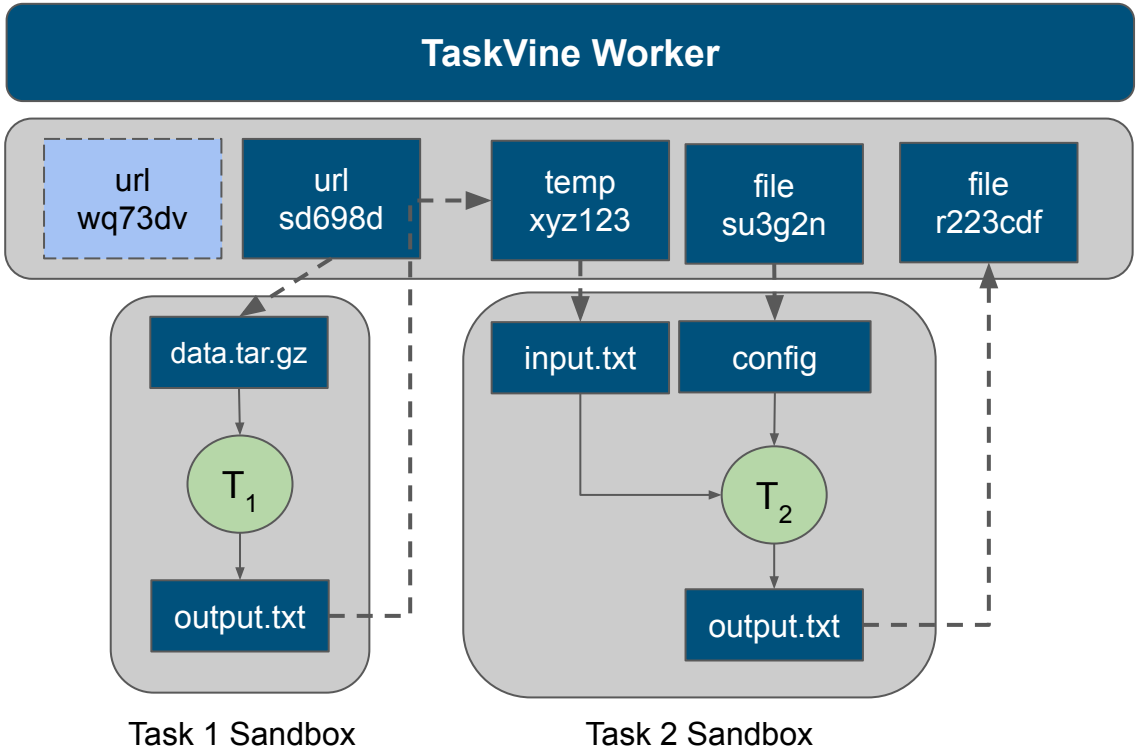
Old App:
Tasks 1-15m

New App: Tasks 30-60s

Runtime Architecture

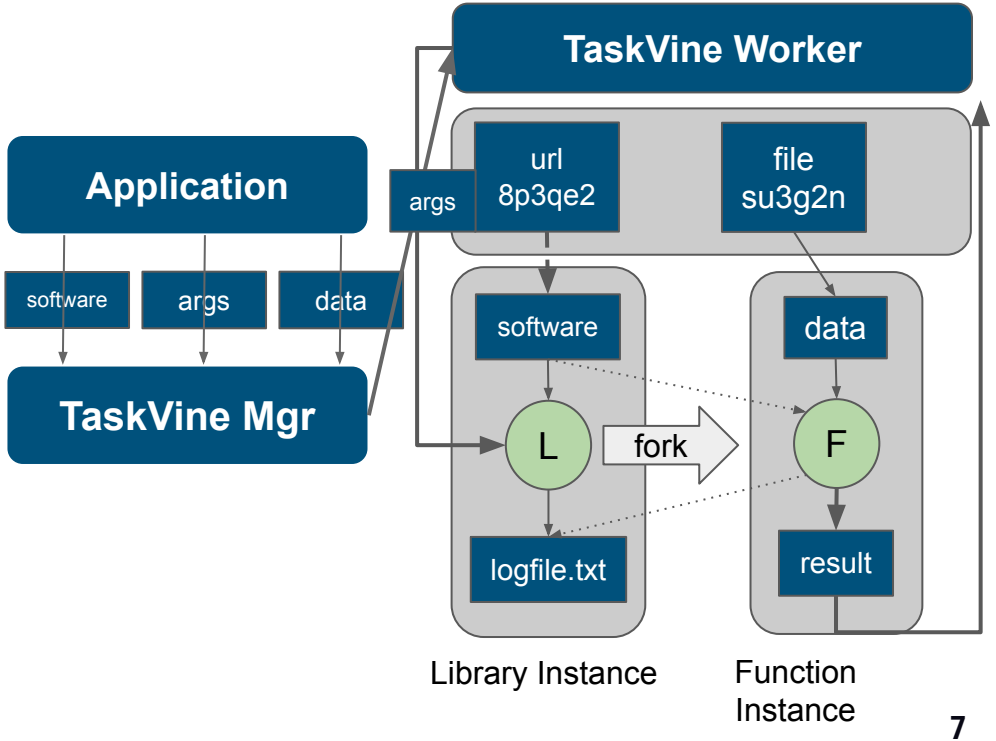


TaskVine: Worker Architecture



Serverless Execution

```
# Define ordinary Python functions
def my_sum(x, y):
    import numpy
    return x+y

def my_mul(x, y):
    import tensorflow
    return x*y

# Create a library object from functions
L = m.create_library_from_functions(
        "my_library", my_sum, my_mul)

# Install the library on all workers.
m.install_library(L)

f = FunctionCall('my_library', 'my_mul', 2, 17)
m.submit(f)
```



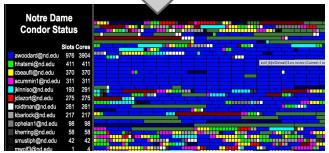
DASK DAG Manager Integration: DaskVine

CCTools

dask

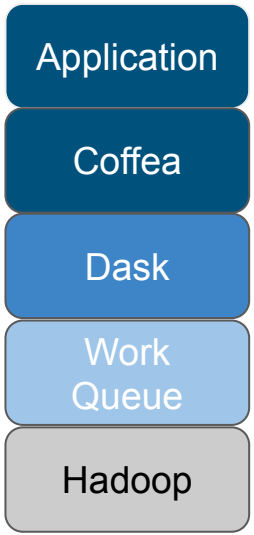
```
# Dask Task Graph
d = {'x': 1,
     'y': (inc, 'x'),
      'z': (add, 'y', 10)}
```

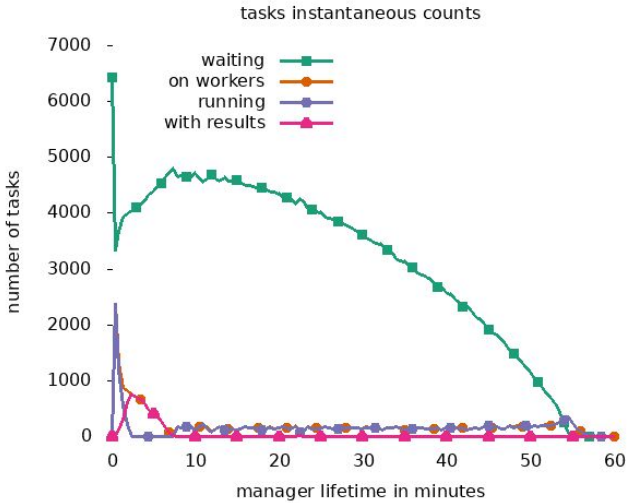
TaskVine

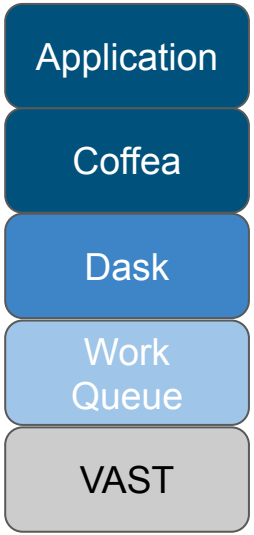


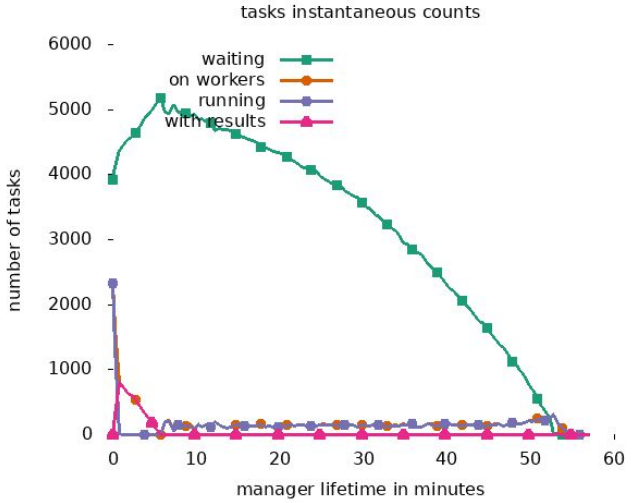

```python
from coffea.nanoevents import NanoEventsFactory
from ndcctools.taskvine import DaskVine

m = DaskVine(name="noncompus-mentis")
events = NanoEventsFactory.from_root(
        {"file:///data//Run2012B_SingleMu.root": "/Events"},
    metadata={"dataset": "SingleMu"}
).events()

q1_hist = (
    hda.Hist.new.Reg(100, 0, 200, name="met", label="$E_{T}^{miss}$ [GeV]")
    .Double()
    .fill(events.MET.pt)
)

q1_hist.compute(
    scheduler=m.get,
    lazy_transfers=True,
    task_mode="function-calls", lib_resources={'cores':12, 'slots':12}
).plot1d()
```

Stack Progression Part 1 (~17500 tasks)



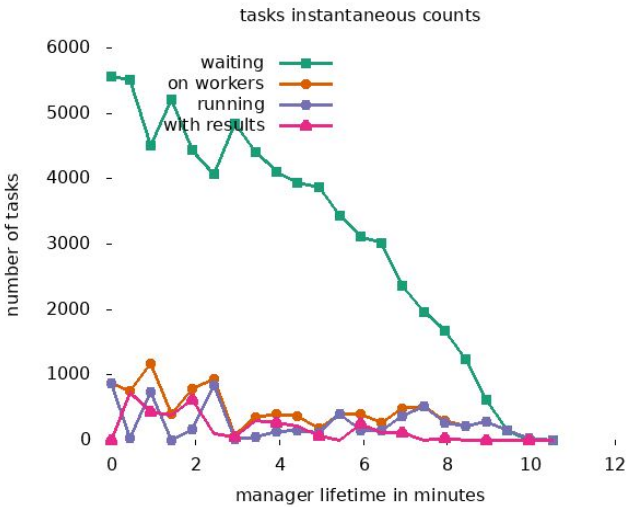




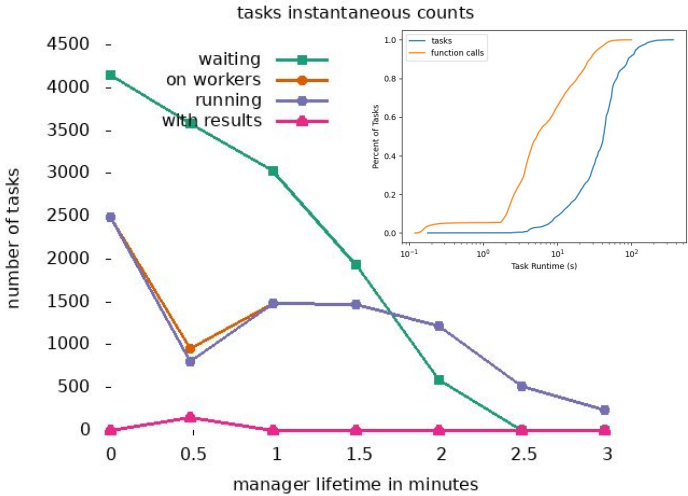


Stack Progression Part 2 (~17500 tasks or calls)

Application
Coffea
Dask
TaskVine
+PythonTask
VAST



Application
Coffea
Dask
TaskVine
+ Functions
VAST



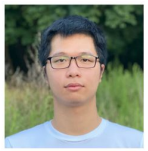
Thanks!

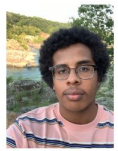
https://cctools.readthedocs.io

https://ccl.cse.nd.edu/software/taskvine


Douglas Thain
Director

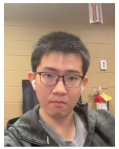

Benjamin Tovar
Research Soft. Engineer


Thanh Son Phung
Ph.D. Student


Barry Sly Delgado
Ph.D. Student


Colin Thomas
Ph.D. Student


Md Saiful Islam
Ph.D. Student


Jin Zhou
Ph.D. Student

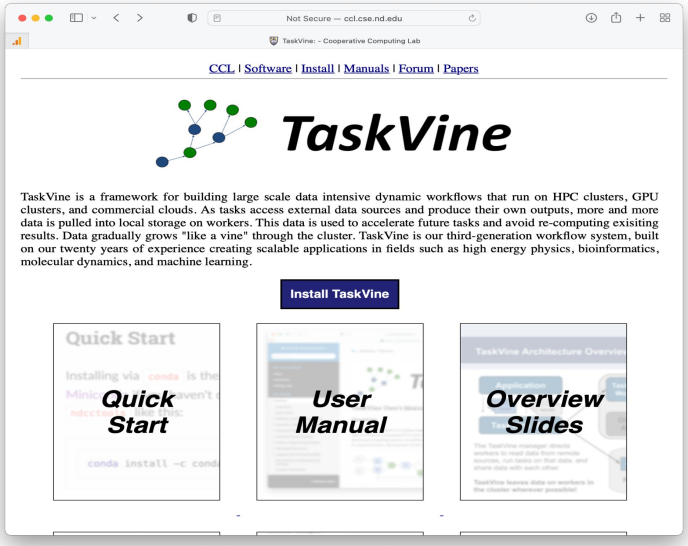

David Simonetti
Undergraduate


Andrew Hennessee
Undergraduate


Jachob Dolak
Undergraduate


Jon Brockett
Undergraduate


Thomas Hieber
Undergraduate

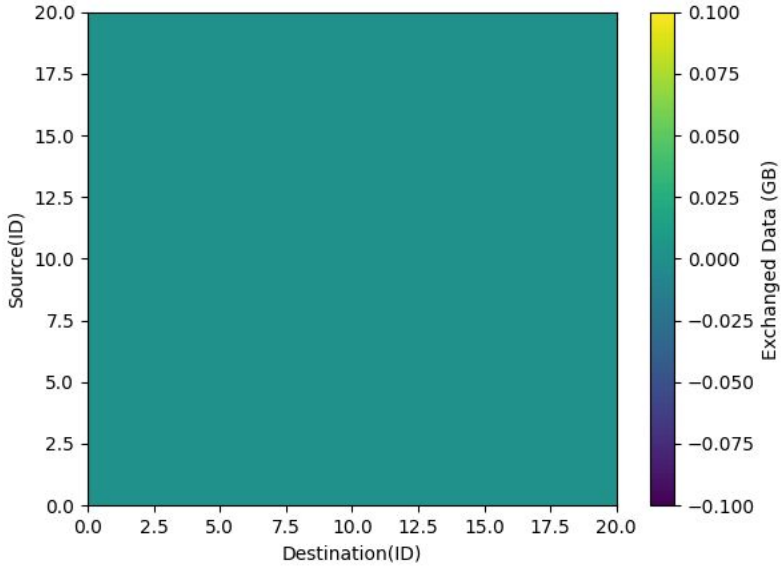


bslydelg@nd.edu

conda install -c conda-forge ndcctools

Extra Slide: Utilizing In-Cluster Bandwidth and Disk (animation not in pdf)

WorkQueue



TaskVine