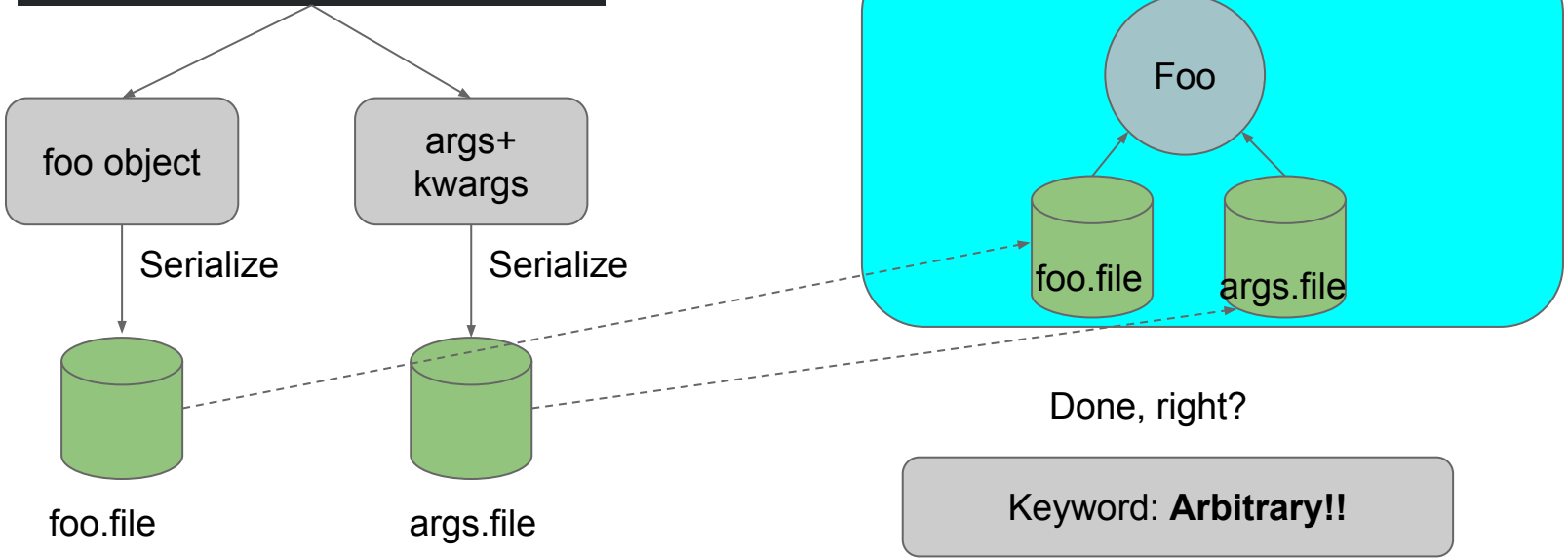
Parsl+TaskVine: Bridging the Data Gap for High-Performance Scientific Workflows in Python

Thanh Son Phung and the CCL Team
University of Notre Dame
ParslFest 2023
Chicago, IL October 2023

Background

Invoke arbitrary Function, ***remotely***

```
def foo(*args, **kwargs):
    ...
```



Done, right?

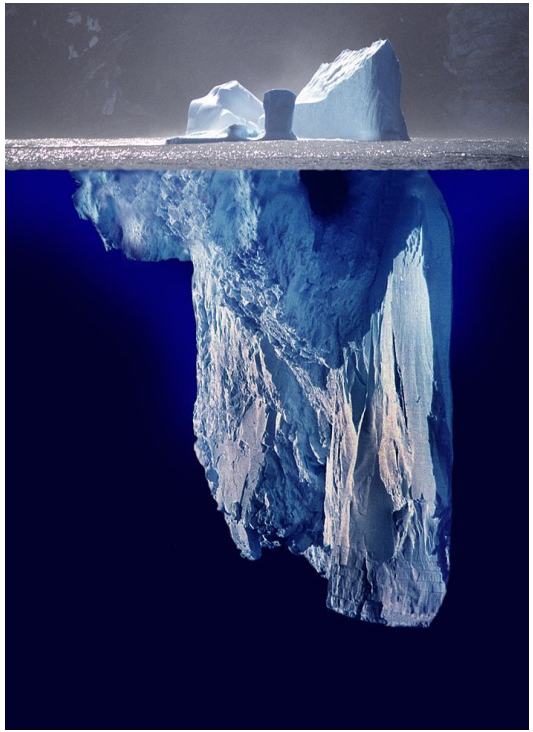
Keyword: **Arbitrary!!**

Background

Invoke ***arbitrary*** Function, remotely

```
def foo(*args, **kwargs):
    ...
```

```
def foo(*args, **kwargs):
    import x
    import y
    from z import a, b, c
    ...
    # input file(s)
    with open('bar', 'r') as f:
        ...
    # output file(s)
    with open('tmp', 'w') as f:
        ...
```



Parsl users see:
- Function object
- args + kwargs

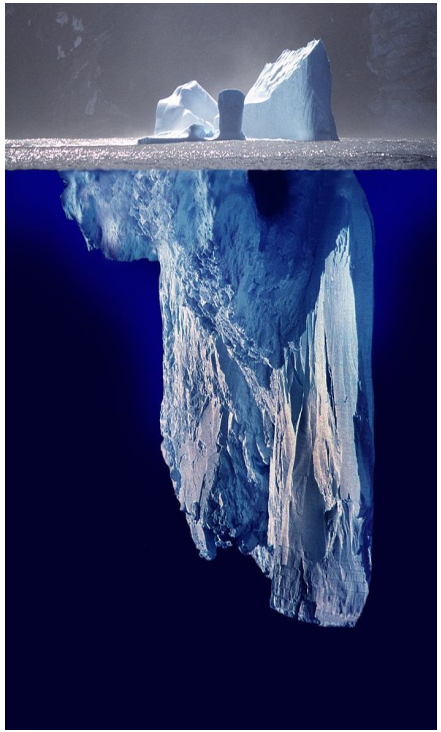
Under-the-hood:
- Package Dependencies
- Input Files
- Output Files
- Intermediate/Temporary Files (function chaining, y = f(g(x)))

Background

Invoke ***arbitrary*** Function,

```
def foo(*args, **kwargs):
    ...
```

```
def foo(*args, **kwargs):
    import x
    import y
    from z import zz
    ...
    # input file(s)
    with open('bar', 'r') as f:
        ...
    # output file(s)
    with open('tmp', 'w') as f:
        ...
```



Parsl users see:
- Function object
- args + kwargs

Under-the-hood
- Package Dependencies
- Input Files
- Output Files
- Intermediate /Temporary Files

(function chaining, y = f(g(x)))

CCTools

Tech Stack

Parsl

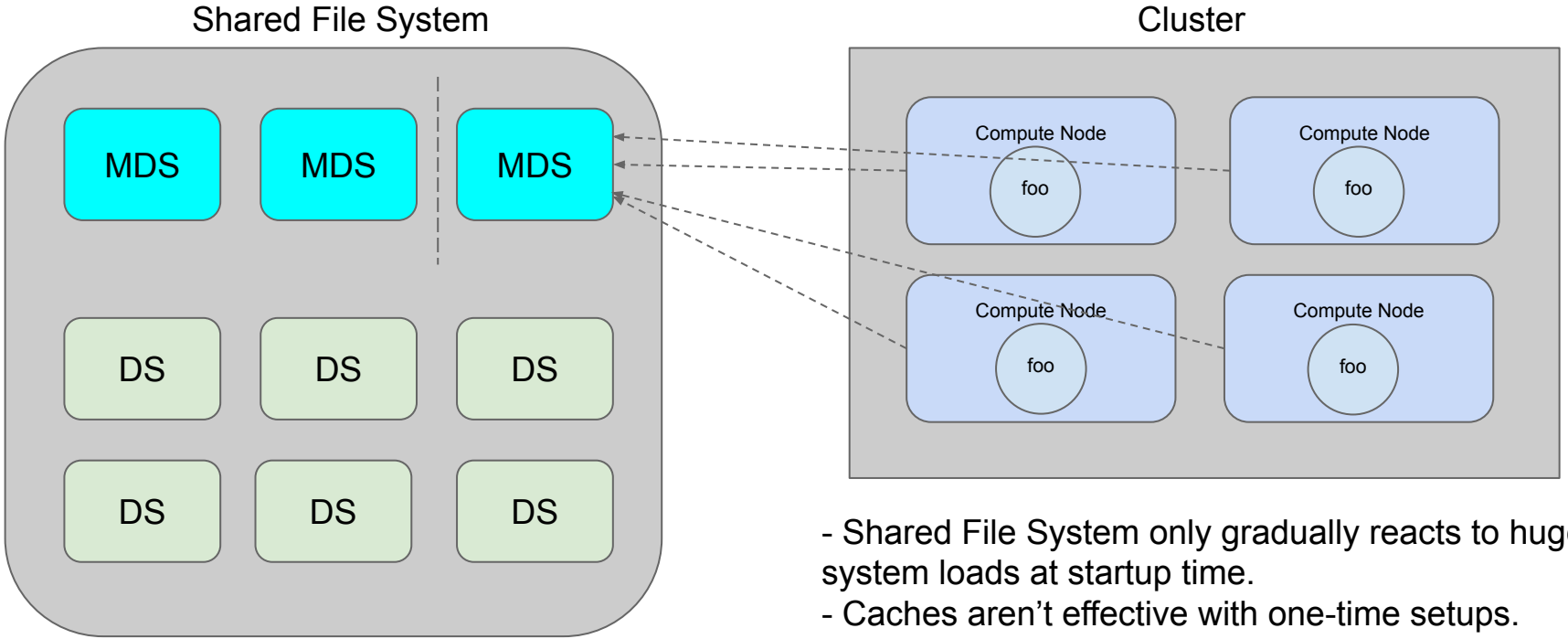
Shared File Systems

lustre

GPFS

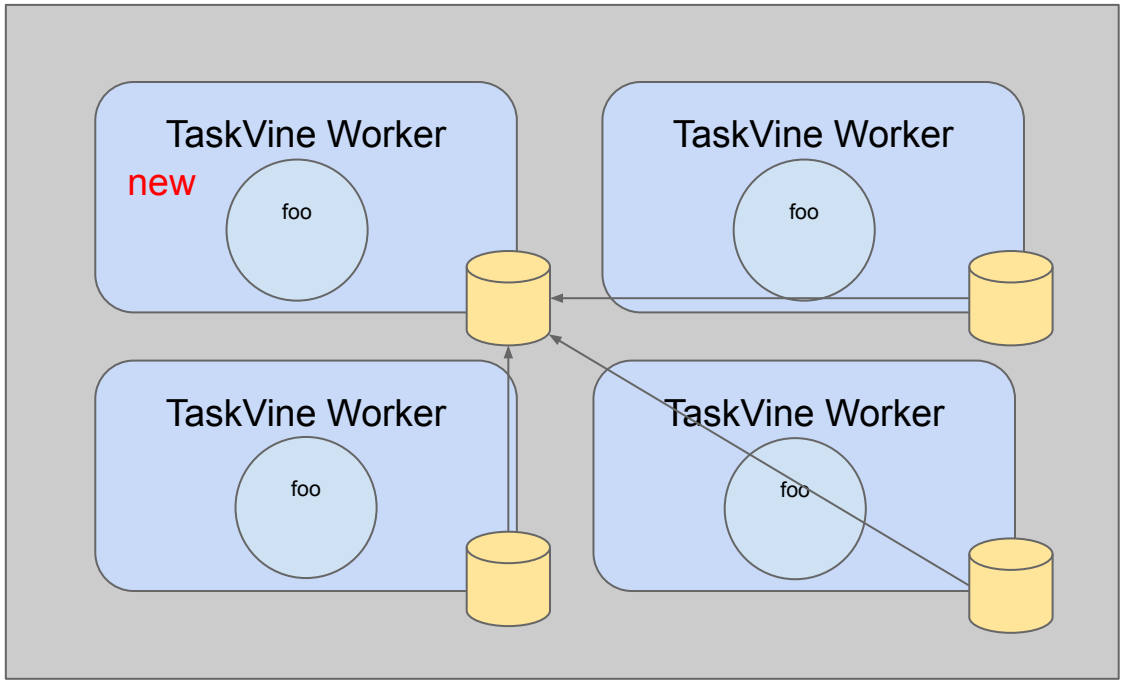
PANASAS

Problem



- Shared File System only gradually reacts to huge system loads at startup time.
- Caches aren’t effective with one-time setups.
- Happens over the course of a workflow execution.

Solution: Parsl+TaskVine



Insights

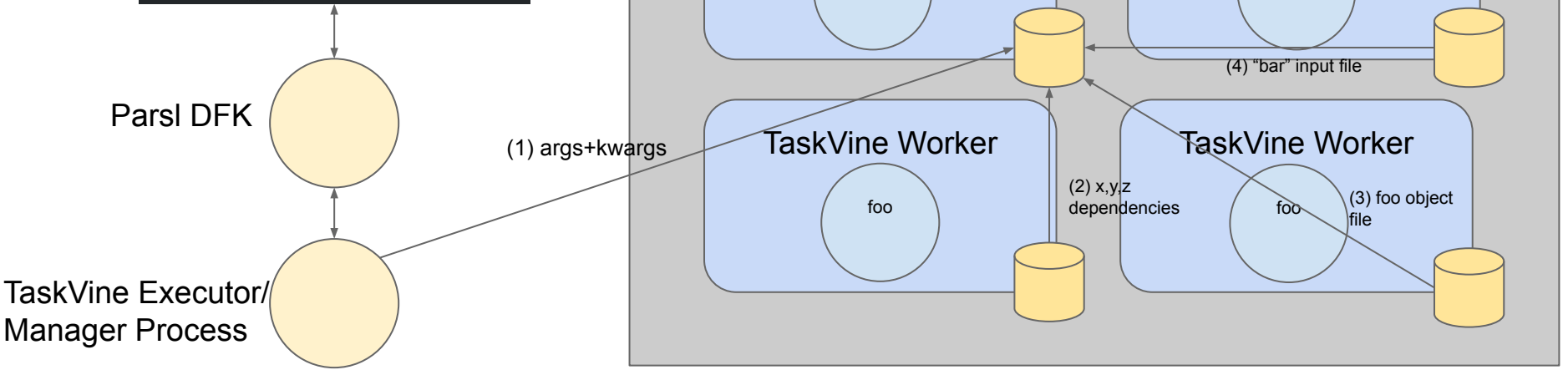
- Costly to move data in and out of cluster.
- Unused local disk storage on compute nodes.
- Data from other nodes aren't used.

TaskVine Novelties

- Explicit Data Placement with Data-to-Task Bindings!
- Data Stays in Cluster until Not Needed.
- No remote I/O! Only local I/O over the course of a workflow execution.
- No shared fs overloading!

Solution: Example

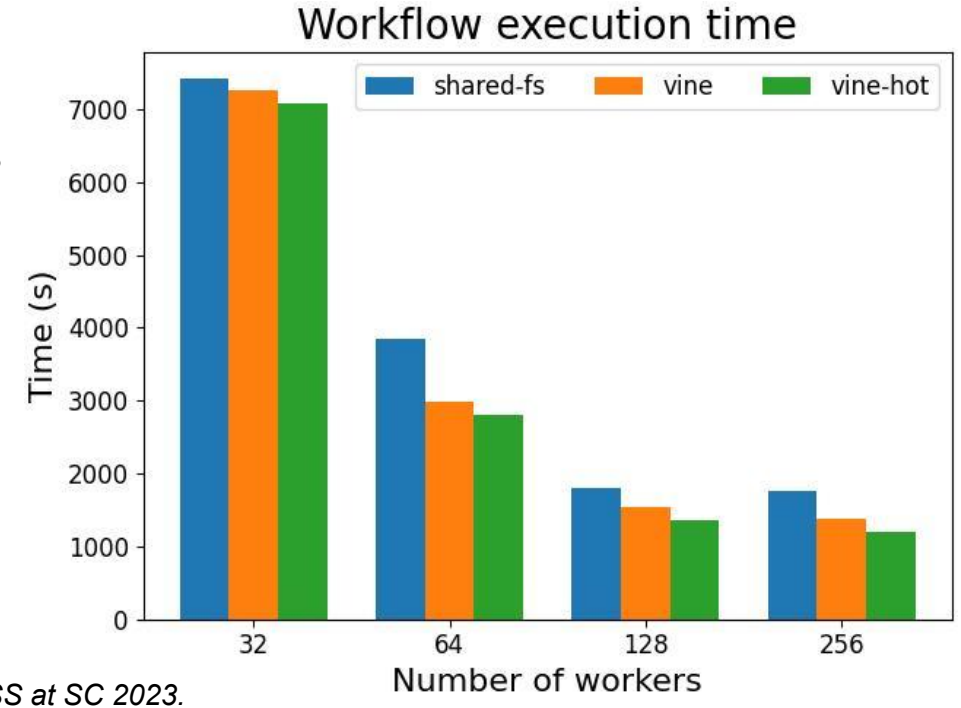
```
def foo(*args, **kwargs):
    import x
    import y
    from z import a, b, c
    ...
    # input file(s)
    with open('bar', 'r') as f:
        ...
    # output file(s)
    with open('tmp', 'w') as f:
        ...
```



Evaluation: Data Distribution Method

Workflow Specification and Setup:
- 2048 function invocations
- Each invocation trains a NN model in 5 minutes.
- Data dependencies include software dependencies (zipped: ~1GB, unzipped: ~3GBs) and dataset (zipped: 17MBs, unzipped: 25MBs).
- Each compute node has 16 cores, 16GBs of memory and disk, and runs 8 tasks concurrently.
- Each worker has access to the local panasas shared file system.
- Varying number of workers -> varying amount of concurrent tasks -> varying pressure on the shared file system.

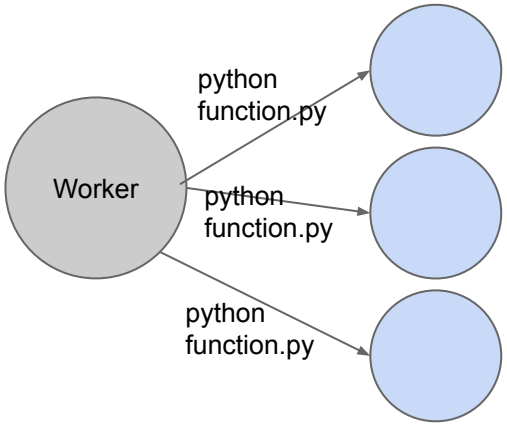
-> TaskVine is more scalable.



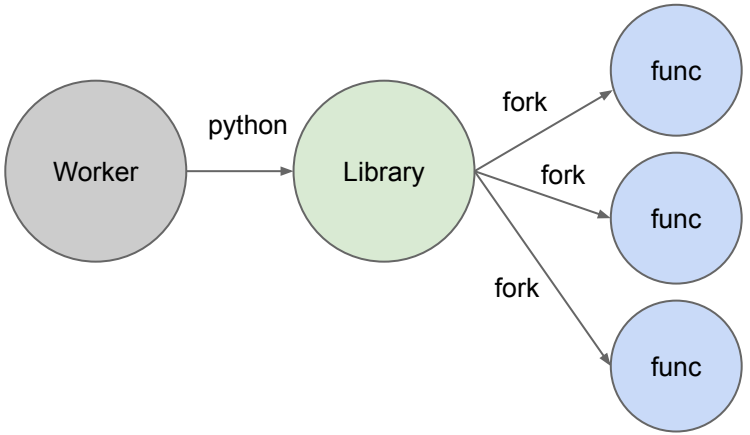
Maximizing Data Utility for HPC Python Workflow Execution, HPPSS at SC 2023.

Optimization: Serverless

Regular Way: Starting a fresh Python interpreter process for every function call.



Serverless: Fork a current process for every function call.



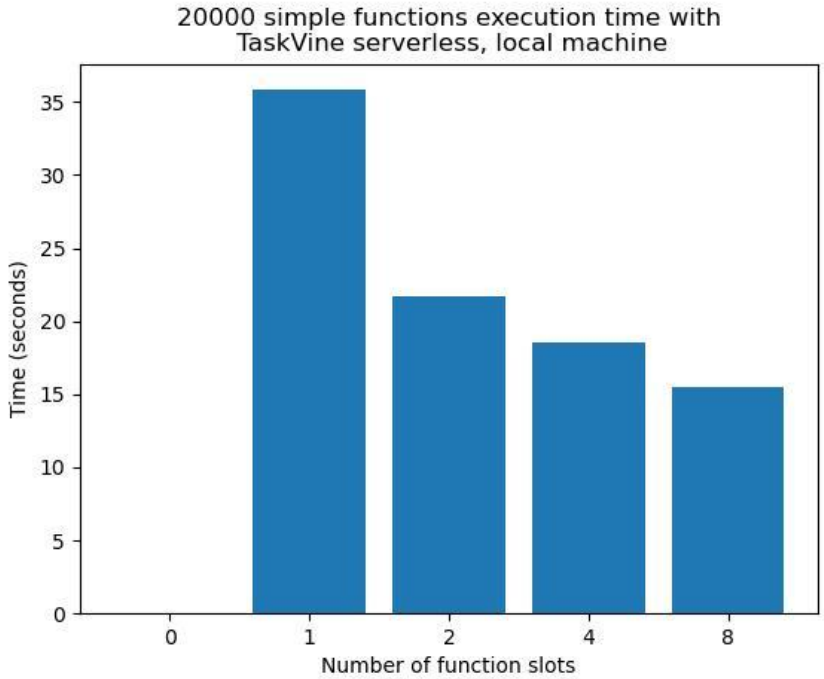
Benefits:

- Don’t invoke fresh interpreter everytime.
- Minimal impact for import statements. (`import tensorflow` takes 5-10 seconds!)
- Function object already in process memory, only loads arguments from file.

Evaluation: Serverless

Run of 20,000 simple functions on a local machine with:
- *regular way* (**711s**, not shown) and
- *serverless* (**~50x speedup**, on right)

-> Great for short-running functions!



TaskVine: Managing In-Cluster Storage for High-Throughput Data Intensive Workflows, *WORKS at SC 2023.*

Try it now!

1) Install via conda (recommended):

```
conda install -c conda-forge ndcctools parsl
```

2) Change configuration of executor:

```
from parsl.config import Config
from parsl.executors.taskvine import TaskVineExecutor, TaskVineManagerConfig
manager_config = TaskVineManagerConfig(project_name='test')
executor = TaskVineExecutor(label='test',
                            worker_launch_method='manual',
                            manager_config=manager_config)
config = Config(executors=[executor])
```

3) More examples: https://cctools.readthedocs.io/en/stable/taskvine/#workflow-integration

Current Status of TaskVine

This work was supported by
NSF Award OAC-1931348

- **TaskVine** is a component of the Cooperative Computing Tools (cctools) from Notre Dame alongside Makeflow, Poncho, Resource Monitor, etc.
- Research software with an engineering process: issues, tests, manual, examples.
- We are eager to collaborate with new users on applications and challenges!

http://cctools.readthedocs.io