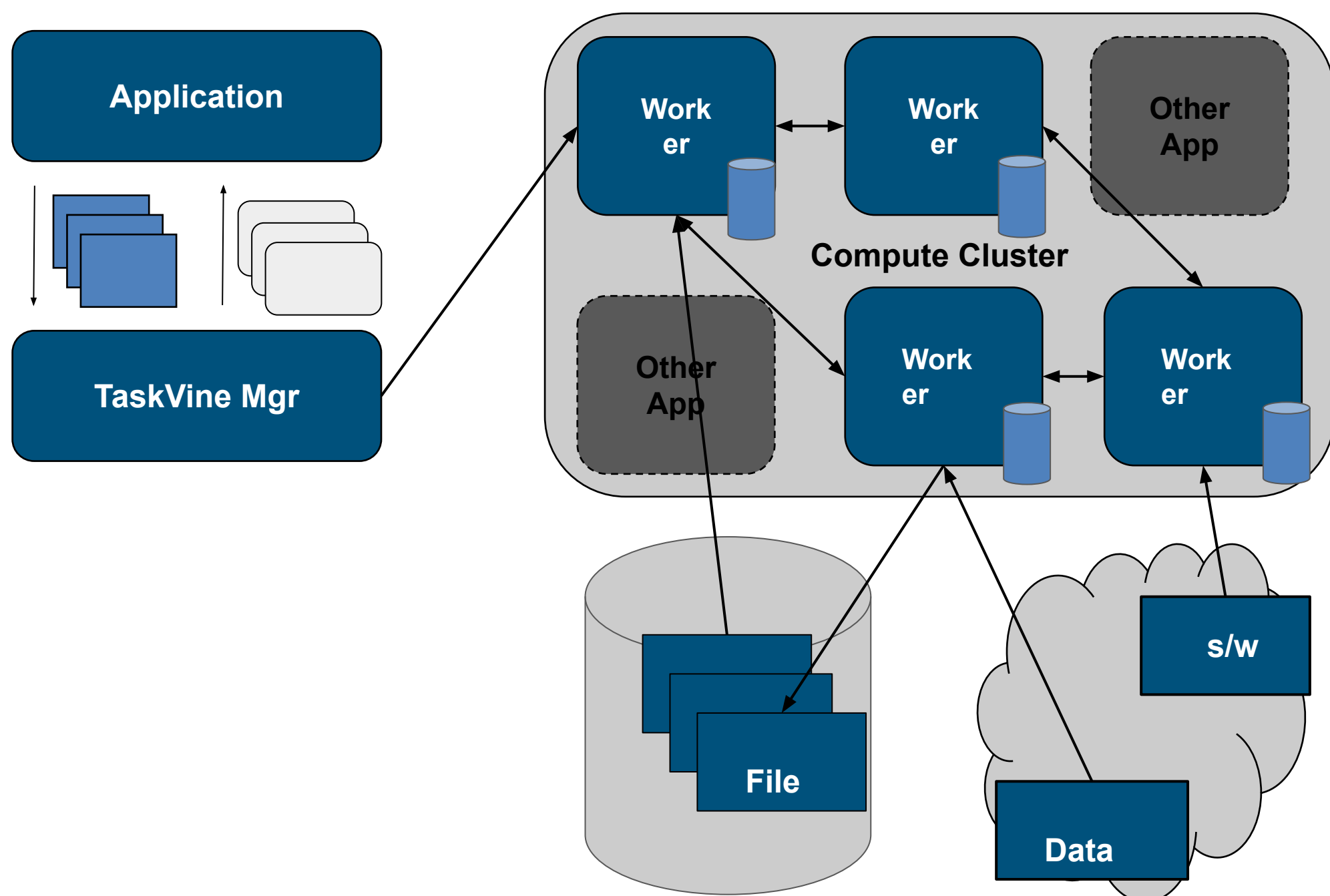


# Import Hoisting for Serverless Libraries

Jin Zhou and Douglas Thain

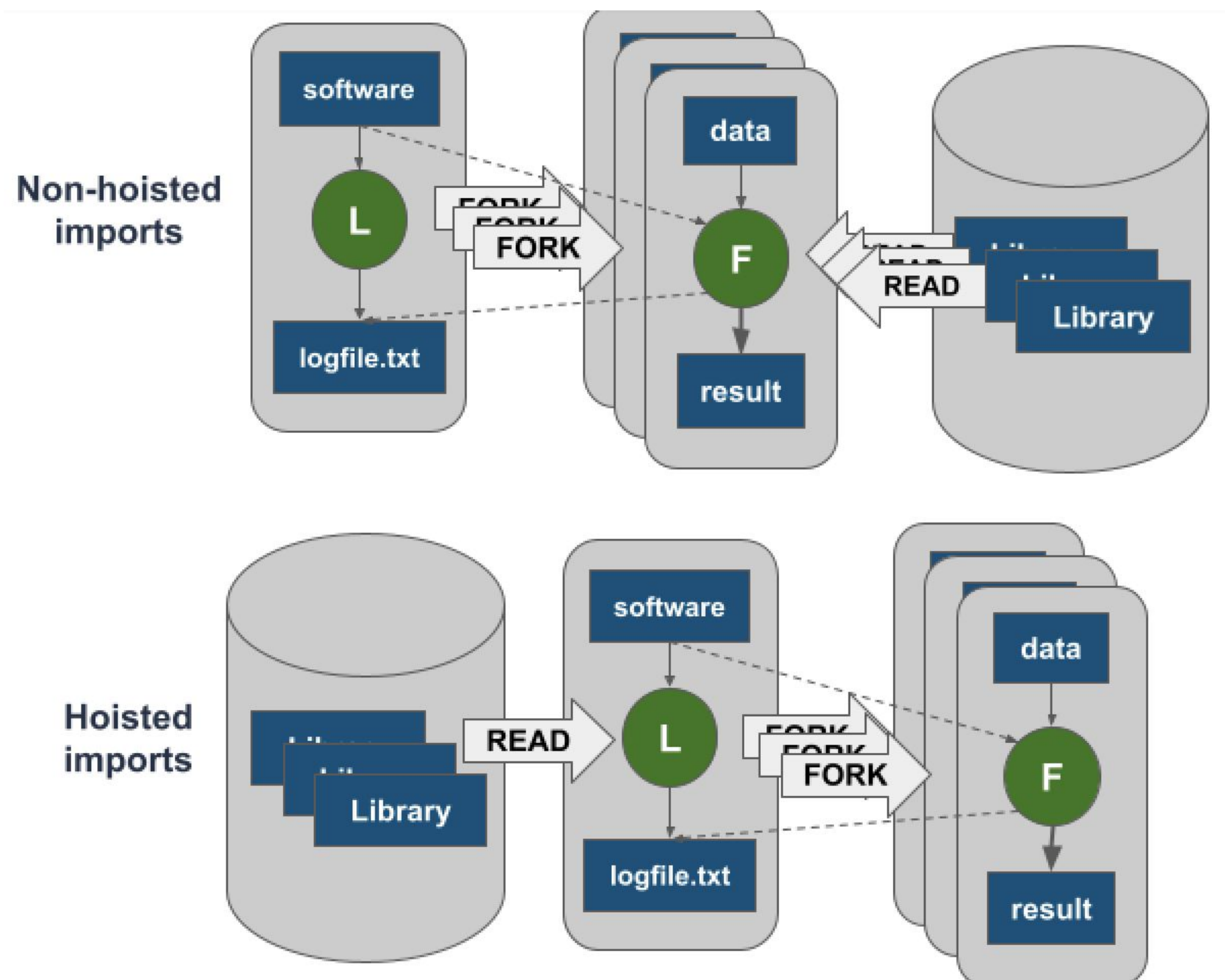
Cooperative Computing Lab, University of Notre Dame  
 {jzhou24, dthain}@nd.edu

**TaskVine** is a system for executing data intensive scientific workflows on clusters, clouds, and grids from very small to massive scale. TaskVine controls the computation and storage capability of a large number of workers, striving to carefully manage, transfer, and re-use data and software wherever possible. The TaskVine manager directs workers to read data from remote sources, run tasks on that data, and share data with each other. **TaskVine leaves data on workers in the cluster wherever possible!**



## Hoisting Imports in Serverless

We optimize task execution in serverless environment by introducing a mechanism to "hoist" import statements to the beginning of library tasks. This approach ensures that specified libraries are loaded only once per library task, rather than at every function call, reducing the overhead associated with disk reads and the loading of large libraries.



## Serverless Computing in TaskVine

TaskVine provides a serverless computing architecture that facilitates running functions reliant on worker-installed libraries, ensuring seamless execution and a user-friendly development experience. This setup involves creating and deploying LibraryTask to define functions and then executing these functions through FunctionCall tasks by referencing the library's name.

```
Import numpy
```

```
libtask = q.create_library('lib', mul_matrix, \
    imports=[numpy])
```

```
task = vine.FunctionCall('lib', 'mul_matrix', m)
```

```
import ndcctools.taskvine as vine
Import numpy

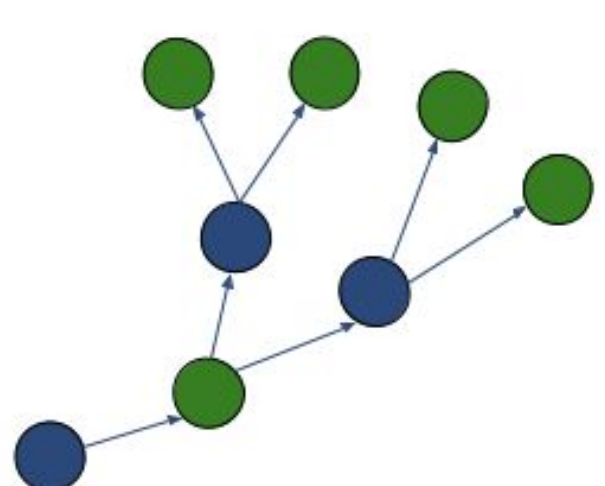
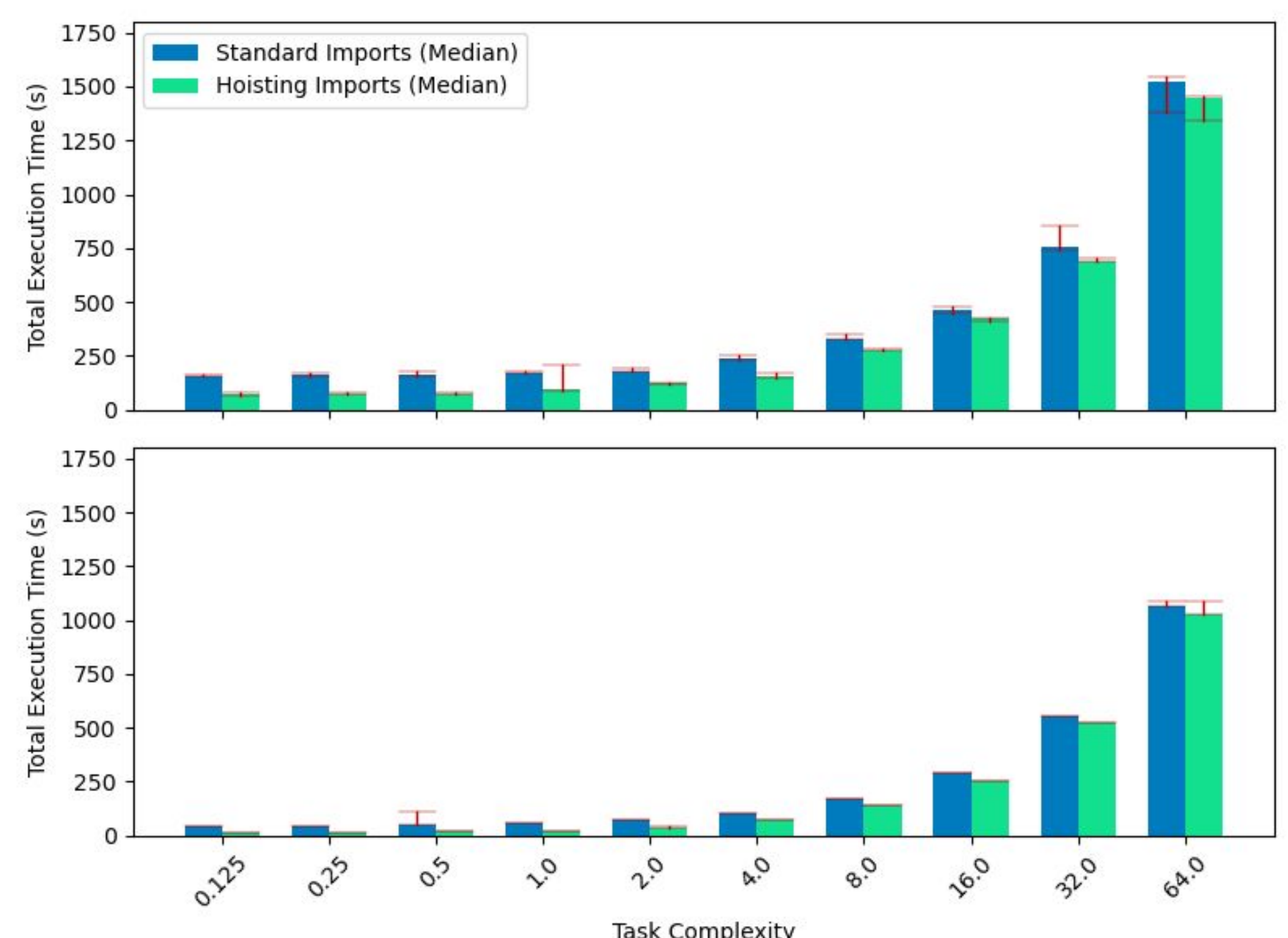
def mul_matrix(matrix):
    import numpy
    return numpy.dot(matrix, matrix)

q = vine.Manager(port=args.port)
libtask = q.create_library('lib', mul_matrix)
q.install_library(libtask)

m = np.random.rand(100, 100)
task = vine.FunctionCall('lib', 'mul_matrix', m)
q.submit(task)
```

## Evaluation

Import hoisting has improvement on both *Shared Filesystem* (top) and *Vine Filesystem* (bottom). Within each case, we gain significant to little benefit as the task complexity grows.



# TaskVine

**Acknowledgement:** This work was supported in part by grant OAC #1931348 "CSSI Elements: Data Swarm: A User-Level Framework for Data Intensive Scientific Computing".