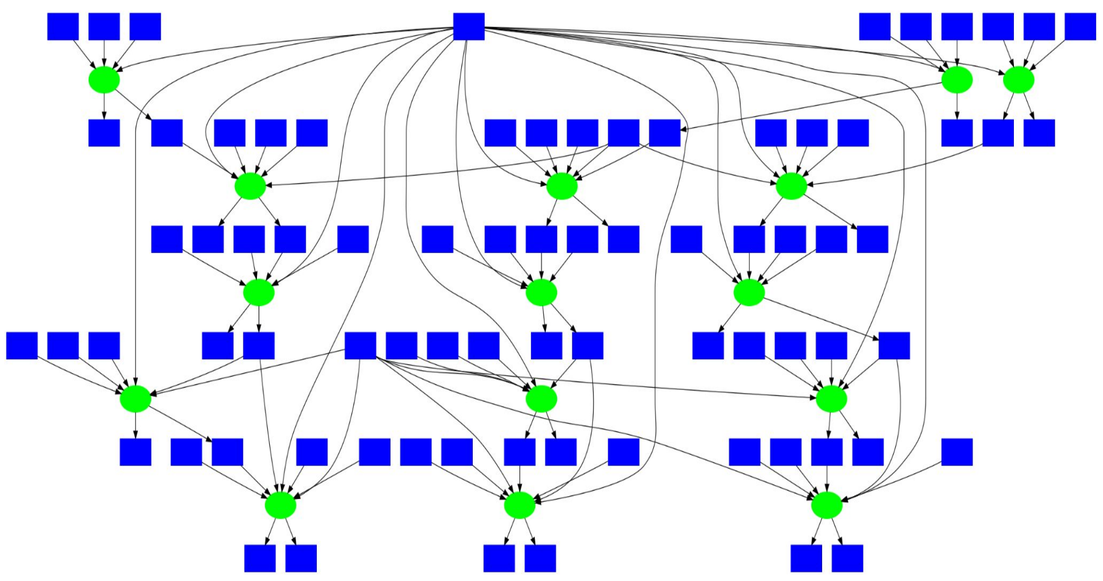
Leveraging Intermediate Data Management with Parsl/TaskVine

Colin Thomas, Douglas Thain, University of Notre Dame
cthoma26@nd.edu

Introduction

Parsl is a parallel scripting library which allows the expression of task-based HPC workflows in Python. The TaskVine executor, developed by Notre Dame's Cooperative Computing Lab, allows Parsl workflows to utilize the local storage of worker nodes, improving cluster bandwidth and minimizing data movement. TaskVine as a standalone workflow system is capable of labeling data dependencies as "temporary files", or intermediate data, which is created in the cluster and used as inputs for subsequent tasks, yet it is not returned to the user.



Pictured above is an ideal workflow DAG. Each task in green consumes and produces a set of files in blue. Many of these tasks are sequentially dependent, and intermediate data is only declared or specified when it is required as an input. However, if a workflow is deployed using shared storage space, the intermediate data will be written and read back from the file system between each task, resembling more of a horizontal DAG as shown below.

Objective

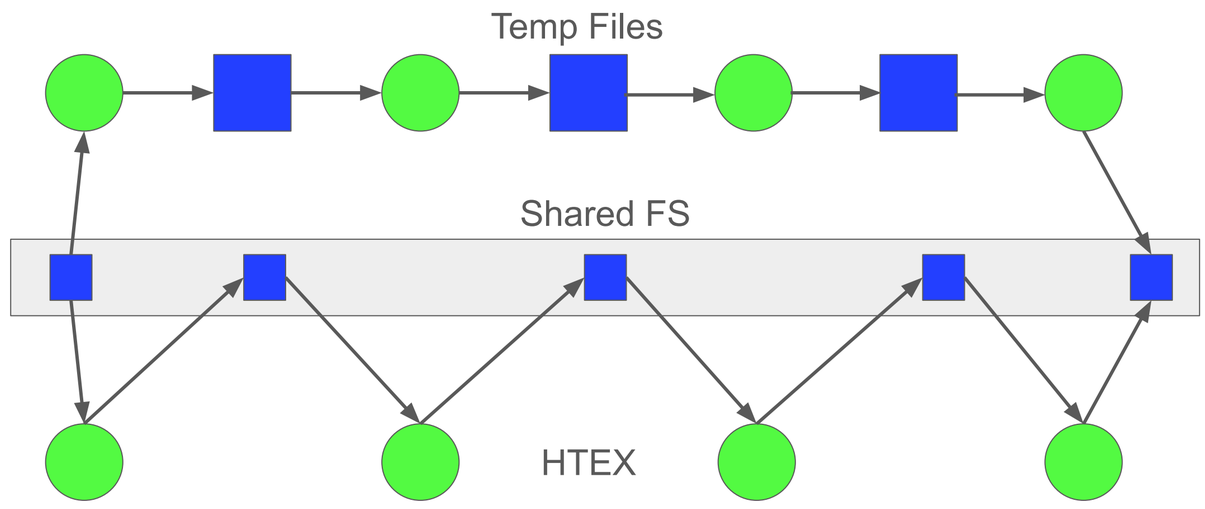
The objective is to extend the structural aspects of the DAG to the movement and access patterns of data dependencies. Data should be created and transferred from one task to another.

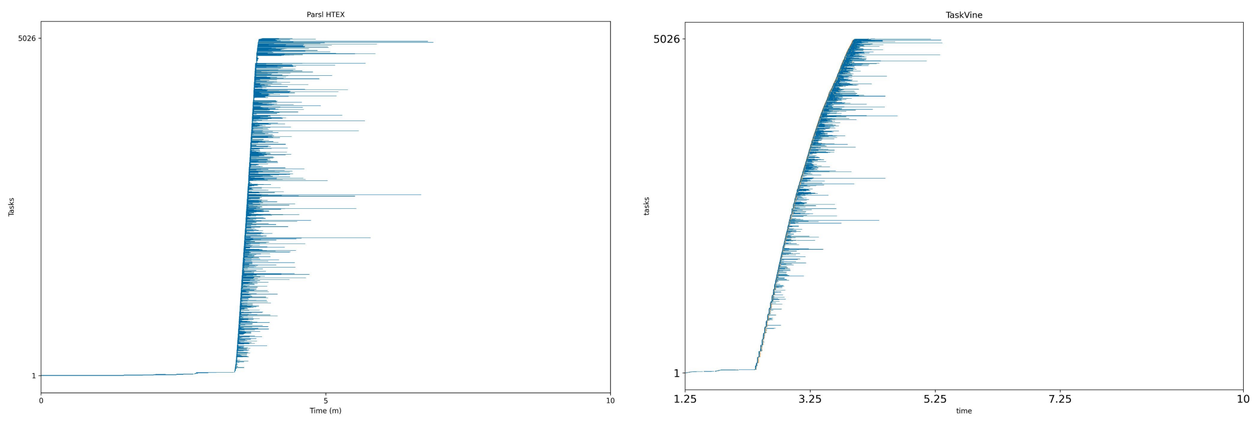
ParslDock

ParslDock is a molecular biology simulation workflow involving batches of sequentially dependent tasks, each task in sequence producing a file as a dependency to the next task. The dag above is one such group of tasks.

```
f_pdb = PFile('taskvinetemp://%s.pdb' % fname)
f_coords_pdb = PFile(f'taskvinetemp://{fname}-coords.pdb')
f_coords_pdbqt = PFile(f'taskvinetemp://{fname}-coords.pdbqt')
f_config = PFile('taskvinetemp://%s-config.txt' % fname)
f_bringback = PFile(f"{fname}-out.pdb")
```

We may now express temporary files in Parsl by using the *taskvinetemp://* protocol. This communicates that the file will be created remotely and will not be returned to the user.





Configuration:	Execution Time:
Parsl/TaskVine temp	5.5 minutes
Parsl/TaskVine base	12 minutes
Parsl HTEX	8 minutes

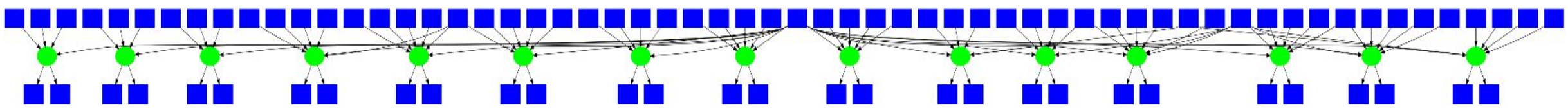
Future Work

Extending temporary files to the Parsl/TaskVine executor offers performance benefits to data-intensive workflows, as well as providing a foundation for future work on fixed location task scheduling.

CCTools

For more information about TaskVine and CCTools visit our documentation → https://cctools.readthedocs.io/en/latest https://ccl.cse.nd.edu/software/taskvine

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



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