**Goals:** Easily write Python workflows that glue together external programs and Python functions; run them quickly and easily in parallel on diverse resources (local, remote, HPC, Cloud, CPU, GPU)

Apps define opportunities for parallelism
- Python apps call Python functions
- Bash apps call external applications

Apps return “futures”: a proxy for a result that might not yet be available

Apps run concurrently respecting data dependencies. Natural parallel programming!

Parsl scripts are independent of where they run.
Write once run anywhere!

Parsl: Productive Parallel Programming in Python


*University of Chicago, "Argonne National Laboratory, "Globus; **Parallel Works
***National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign

Award #: 1550476, 1550475, 1550528, 1550562, 1550588

NSF CSSI PI Meeting, Seattle, WA, Feb. 13-14, 2020

http://parsl-project.org

```python
@python_app
def hello():
    return 'Hello World!'

print(hello().result())
```

```bash
@bash_app
def echo_hello(stdout='echo-hello.stdout'): return 'echo "Hello World!"'

echo_hello().result()

with open('echo-hello.stdout', 'r') as f:
    print(f.read())
```

Hello World!