Argonne NATIONAL LABORATORY

On-demand Scientific Services: AlphaFold

funcX

Ryan Chard

globus labs

funcX interface for on-demand HPC services

Leadership computing facilities:

- Extreme scale
- Specialized hardware
- Enormous datasets (simulation, reference, ML)

Also…

- Restrictive queueing models
- Strict authentication and authorization
- Auditing and reporting requirements
- Challenging environments

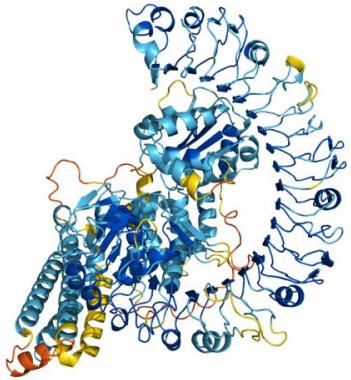
funcX can simplify access to data and compute for many communities

AlphaFold as a Service at ALCF

Cutting edge ML technique to predict protein structure with applications in screening, therapeutics, light sources, crystallography, etc.

```
>GA98_DM.114 GA98 Deletion Mutation Sequence
TTYKLILNLKQAKEEAIKELVDAGTAKYFKLIANAKTVEGVWTLKDE
```

Deployed funcX to run AlphaFold on-demand on ALCF resources



AlphaFold as a Service:

1. User provides FASTA input
2. Upload data to ALCF's Eagle storage system
3. funcX provisions GPU resources on Polaris
4. Task runs for ~1.5 hours
5. Results made available via Globus
6. Download link emailed to user

funcX + AlphaFold + ALCF

funcX endpoint deployed on Polarislogin1

- Shared with private Globus group
  - Restricted to ALCF users
- Singularity-enabled
- Provisions GPU nodes
- Mounts necessary data
- Uses preemptable queue
- Runs in my account

Custom Singularity container

- https://github.com/hyoo/alphafold_singularity

Defined and registered a function to invoke alphafold in container

Also deployed a Globus flow to use the funcx function, runnable by the group

```
config = Config(
    executors=[
        HighThroughputExecutor(
            max_workers_per_node=1,
            strategy=SimpleStrategy(max_idletime=60
            address=address_by_hostname(),
            scheduler_mode='soft',
            worker_mode='singularity_reuse',
            container_type='singularity',
            container_cmd_options="--nv -H /home/ro
            provider=PBSProProvider(
                cpus_per_node=32,
                select_options="ngpus=4",
                launcher=SingleNodeLauncher(),
                account='APSDataAnalysis',
                queue='preemptable',
                scheduler_options=user_opts['polari
                worker_init=user_opts['polaris']['w
                walltime='06:00:00',
                nodes_per_block=1,
                init_blocks=0,
                min_blocks=0,
                max_blocks=4,
            ),
        )
    ],
)
```

Using it

Run via funcX or Globus Flow

- CLI: https://github.com/globus-labs/globus-alphafold-cli

```
$ python cli.py run --fasta /path/to/file.fasta
```

- Or run the flow directly ->

Results emailed with download link:

Globus AlphaFold flow completed.

You can collect the result here: https://g-719d9.fd635.8443.data.globus.org/output/517e5fd2/GB98_DM_3.fasta.log

```
flow_id = '7c277b80-2cca-42b7-a75d-a970841ee874'
flow_scope = 'https://auth.globus.org/scopes/7c277b80-2

flow_input = {
    "input": {
        "fasta": fasta,
        "email": email_address,
    }
}
```

Now start the flow. We create a `flows_client` and will then be prompt

```
flows_client = create_flows_client()

flow_action = flows_client.run_flow(flow_id, flow_scope
flow_action_id = flow_action['action_id']
print(f"flow started: https://app.globus.org/runs/{flow
```

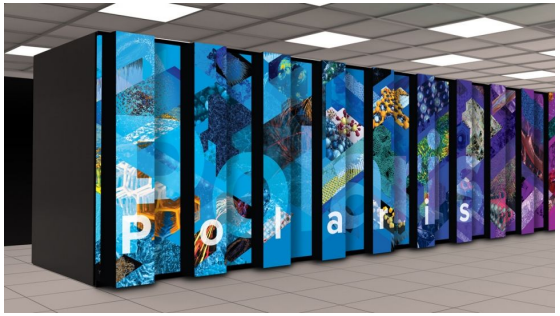
```
Please log into Globus here:
------------------------------
https://auth.globus.org/v2/oauth2/authorize?client_id=e
Fauth.globus.org%2Fv2%2Fweb%2Fauth-code&scope=https%3A%
dffcfe%2Fflow_f616749e_ef70_4991_9429_dd803fdffcfe_user
fRa_KRhBGACrTYmyj16xASTXU8vnw5c&code_challenge_method=S
mmand+Line+Interface+on+mbp.lan
------------------------------

Enter the resulting Authorization Code here: TYWzm5mQox
flow started: https://app.globus.org/runs/cadfa395-30b4
```

Check the status of the flow. The AlphaFold step often takes roughly two

```
flow_action = flows_client.flow_action_status(flow_id,
flow_status = flow_action['status']
print(f'Flow status: {flow_status}')
```

```
Flow status: ACTIVE
```

Production services with a funcX interface



Service account to operate endpoint and multiplex users
- Globus app credentials to own the endpoint

Project allocation to monitor and manage resource usage

Managing functions - vet, add, and remove functions

Manage user access - Globus group membership

Fine-grained auditing and reporting

Mechanism to stop bad tasks without impacting other users

Persist endpoints and restart as necessary (failures, maintenance)
- Service node to operate endpoint outside login nodes
- Considering podman for endpoint management

Ability to update execution environment and container

Next steps

ALCF	funcX	AlphaFold service
Define function vetting process	Improve public auditing capabilities	Provide a web interface
Work through service accounts	Enhance function authorization/ management for endpoints	Leverage in light source analysis loop with APS's GM/CA beamline
Persist endpoints on service nodes via systemd or similar	Task cancellation and management	Integrate into ML training workflows

Thanks!

rchard@anl.gov