Flywheel HPC Client engine installation

Version 1.0.0

The HPC Client is a self-service solution that allows Flywheel jobs and gears to run in a High-Performance Computing environment. Use on-premise hardware that's already available for highly-concurrent scientific workloads. When setting up an HPC cluster, use our "example/settings" as shown here:

https://github.com/flywheel-io/hpc-client/tree/master/examples/settings

1. Flywheel-related settings are stored in the `credentials.sh` file, including the drone secret provided by the Flywheel team.

2. Settings for the cluster where the jobs will run are stored in the cast.yaml.

3. After receiving the **'engine'** binary from the Flywheel Team (e.g., engine--2021-05-20--15.5.1), place it in the following directory:

   `fw-cast/logs/temp`.

4. Create a symlink to the engine binaries so they can be updated in the future without disturbing jobs in flight. Make sure your working directory is the temp directory where the binaries were just stored.

   ```
   ln -s <engine_binaries_name> engine
   ```

   You can check if the link was created with `ls -lah`, which will show something like the following:

   ```
   engine -> engine--2021-05-20--15.5.1
   ```

5. Ensure the engine binary is executable with the command

   ```
   chmod +x <engine_binaries_name>
   ```

6. The `start.sh` script gets settings from both the cast and credentials files. After startup, Flywheel is polled for work marked for the cluster.

7. When work is discovered from Flywheel, the python in the code/cluster directory will format a bash script from a template that will submit a job formatted to the specifications of the scheduler for your particular cluster. An example template is at the end of this file:

   https://github.com/flywheel-io/hpc-client/blob/master/code/cluster/slurm.py

8. After the work has been submitted to the queue, the engine will both monitor the job status in the HPC and manage the upload of results to Flywheel when the work is complete.