Quick jobs

Quick job is one of our web portal's features. It is an excellent way to unify and speed up your team's workflow. For example, you may define what computing resources are required, what software to use, and where the output goes. You may also expose options to your teammate to fine-tune an individual run.

Quick jobs are typical .sbatch scripts. The portal will open a job launcher window when one clicks a .sbatch file. Then you customize the launcher behavior by optional metadata.

Below is a multi-GPU deep reinforcement learning task with a custom description and several exposed options.

```
#!/usr/bin/env bash


#SBATCH -J sac
#SBATCH -o sac.out
#SBATCH -p gpu
#SBATCH -n 8
#SBATCH -N 1
#SBATCH -c 4
#SBATCH --gpus-per-task a100:1
#SBATCH --mem-per-cpu=16000

<<setup
desc: Train a Soft Actor Critic (sac) model on OpenAI Gym environments.
inputs:
  - code: env_id
    display: Environment Id
    type: dropdown
    default: BipedalWalker-v3
    options:
      - BipedalWalker-v3
      - LunarLanderContinuous-v2
      - AntBulletEnv-v0
      - InvertedPendulumBulletEnv-v0
      - CartPoleContinuousBulletEnv-v0
      - PongNoFrameskip-v4
```

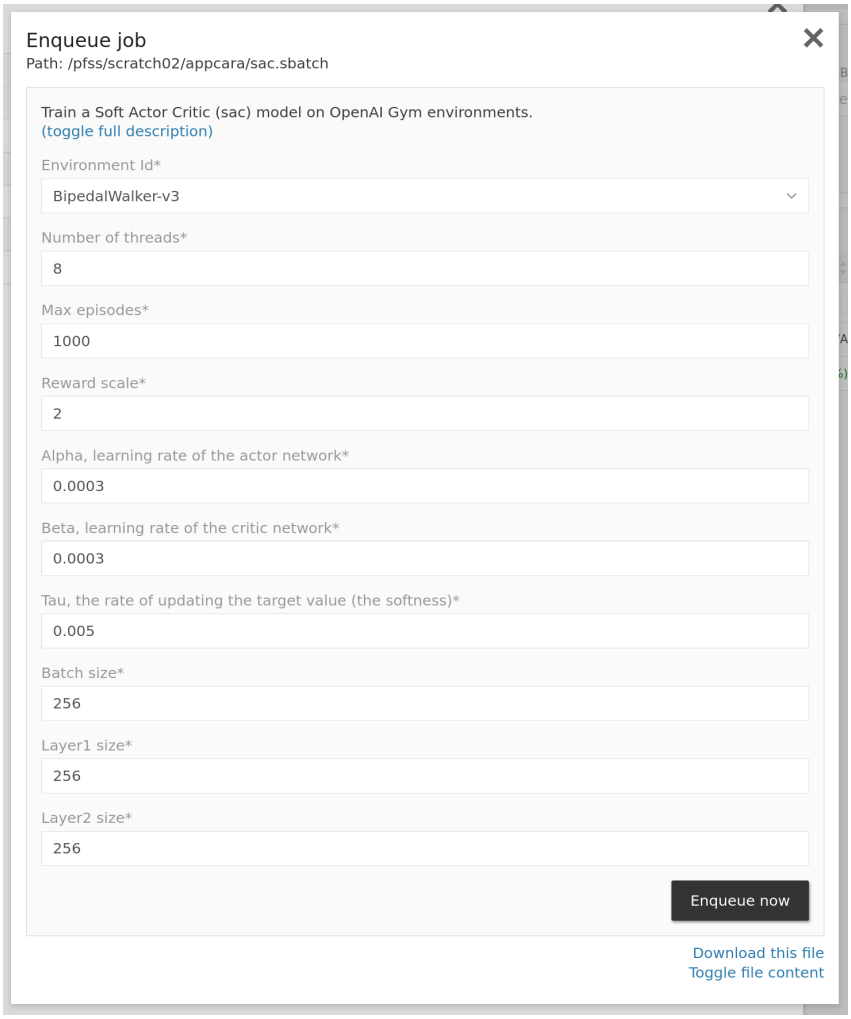
```
  required: true
- code: num_threads
  display: Number of threads
  type: text
  default: 8
  required: true
- code: max_episodes
  display: Max episodes
  type: text
  default: 1000
  required: true
- code: reward_scale
  display: Reward scale
  type: text
  default: 2
  required: true
- code: alpha
  display: Alpha, learning rate of the actor network
  type: text
  default: 0.0003
  required: true
- code: beta
  display: Beta, learning rate of the critic network
  type: text
  default: 0.0003
  required: true
- code: tau
  display: Tau, the rate of updating the target value (the softness)
  type: text
  default: 0.005
  required: true
- code: batch_size
  display: Batch size
  type: text
  default: 256
  required: true
- code: layer1_size
  display: Layer1 size
  type: text
  default: 256
```

```
    required: true
  - code: layer2_size
    display: Layer2 size
    type: text
    default: 256
    required: true
extra_desc: |+
  output model will be stored in ./sac_model
  loss for each episode is plotted to ./sac_loss.png
setup

module load GCC/11.3.0 OpenMPI/4.1.4

mpiexec singularity exec --nv \
  --env env_id=%env_id% \
  --env num_threads=%num_threads% \
  --env max_episodes=%max_episodes% \
  --env reward_scale=%reward_scale% \
  --env alpha=%alpha% \
  --env beta=%beta% \
  --env tau=%tau% \
  --env batch_size=%batch_size% \
  --env layer1_size=%layer1_size% \
  --env layer2_size=%layer2_size% \
  /pfss/scratch02/appcara/gym.sif python sac.py
```

There is a section colored orange that defines how it interfaces with users. It is in YAML format and consists of three sub-sections: desc, inputs, and optional extra_desc. The launcher renders the below form to capture the user's input. The system expects placeholders in format %input_code% in the file content and will replace them with the user input.



The built-in quick jobs

The three built-in quick jobs discussed in the previous section are also made of the above syntax. You may find them in the parallel file system.

- Jupyter Lab in /pfss/toolkit/start_jupyter.sbatch
- VNC in /pfss/toolkit/start_vnc.sbatch
- Run container in /pfss/toolkit/run_container.sbatch

Revision #4
Created 13 November 2022 16:25:43 by Loki Ng
Updated 14 November 2022 02:07:30 by Loki Ng