Deep Q-Learning for RL4Sys Framework

Jeffrey Wang, UNC Charlotte
Dong Dai, Department of Computer Science

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Introduction

- Need for a framework which is able to supply Deep RL models, allowing others to simply connect their current system code with effective and efficient RL algorithms
- Based off of the RL4Sys framework [1], which aims to provide this prewritten code in a flexible system
- Framework originally relied on PPO advantage actor-critic model

Objectives

- Extend the functionality of the RL4Sys framework to provide users with multiple options for reinforcement learning models
- Generalize structure of framework to enable further extensibility with different models
- Demonstrate performance of Deep Q-Learning network against original PPO model

Collected Data

- DQN model was run against synthetic and real-world job traces

Visualization

- average bounded slowdown
- average waiting time
- average turnaround time
- resource utilization
- average slowdown

Method

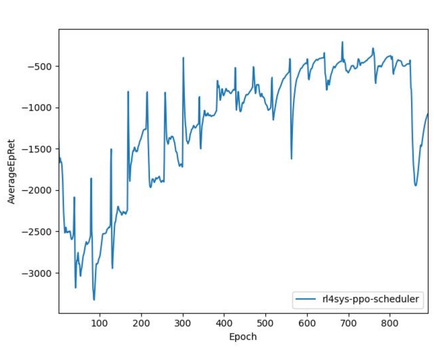
Q-Learning

- RL4Sys uses a **server and client** to enable off-machine training of the deep RL model [1]
- As a result, we use hard-updates to the Q-learning model.

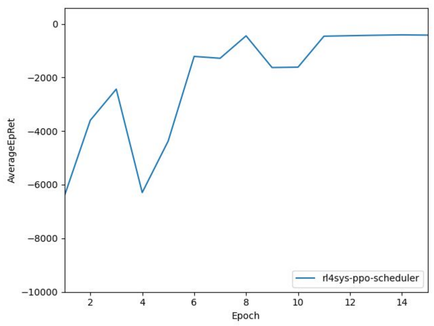
Design

- Framework parts were rewritten to ease **environment writing** and **compartmentalization**
  - Agent, Action data
  - Split training server
  - Documentation

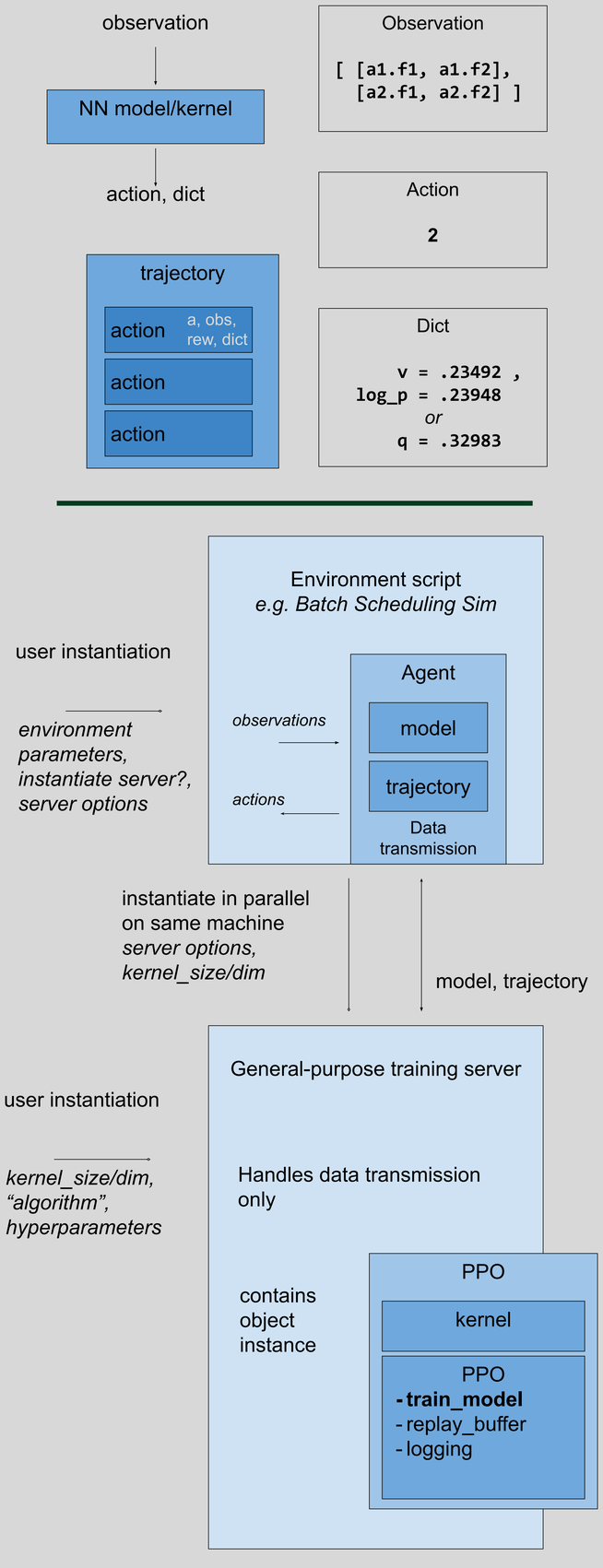
Results



lublin_256.swf



SDSC-SP2



Conclusions

- Deep Q-Learning model less performant than original PPO code
- Future extensibility to framework algorithms and environment examples
- Soft-update design for DQN model

References

[1] Dai et. al. DIR-LAB RL4Sys GitHub repository. github.com/DIR-LAB/RL4Sys/ *Accessed March 2024.*
[2] Paszke and Towers, Reinforcement Learning (DQN) Tutorial. *PyTorch Documentation. Accessed March 2024.*