Overcoming RL Limitations in HPC Scheduling: A Model-Based MCTS Approach for Practical Deployment

Yash Kurkure, Yihe Zhang, Michael E. Papka, Zhiling Lan

Motivation

High-performance computing (HPC) job scheduling has seen promising advances with Deep Reinforcement Learning (DRL). However, challenges such as low interpretability, instability, and high computational cost hinder DRL's practical adoption. We explore a model-based alternative using Monte Carlo Tree Search (MCTS) to overcome these limitations. By leveraging existing HPC simulators as models for MCTS and focusing on transparent decision-making, we aim to develop a scalable and interpretable scheduling solution fit for real-world deployment.

DRL

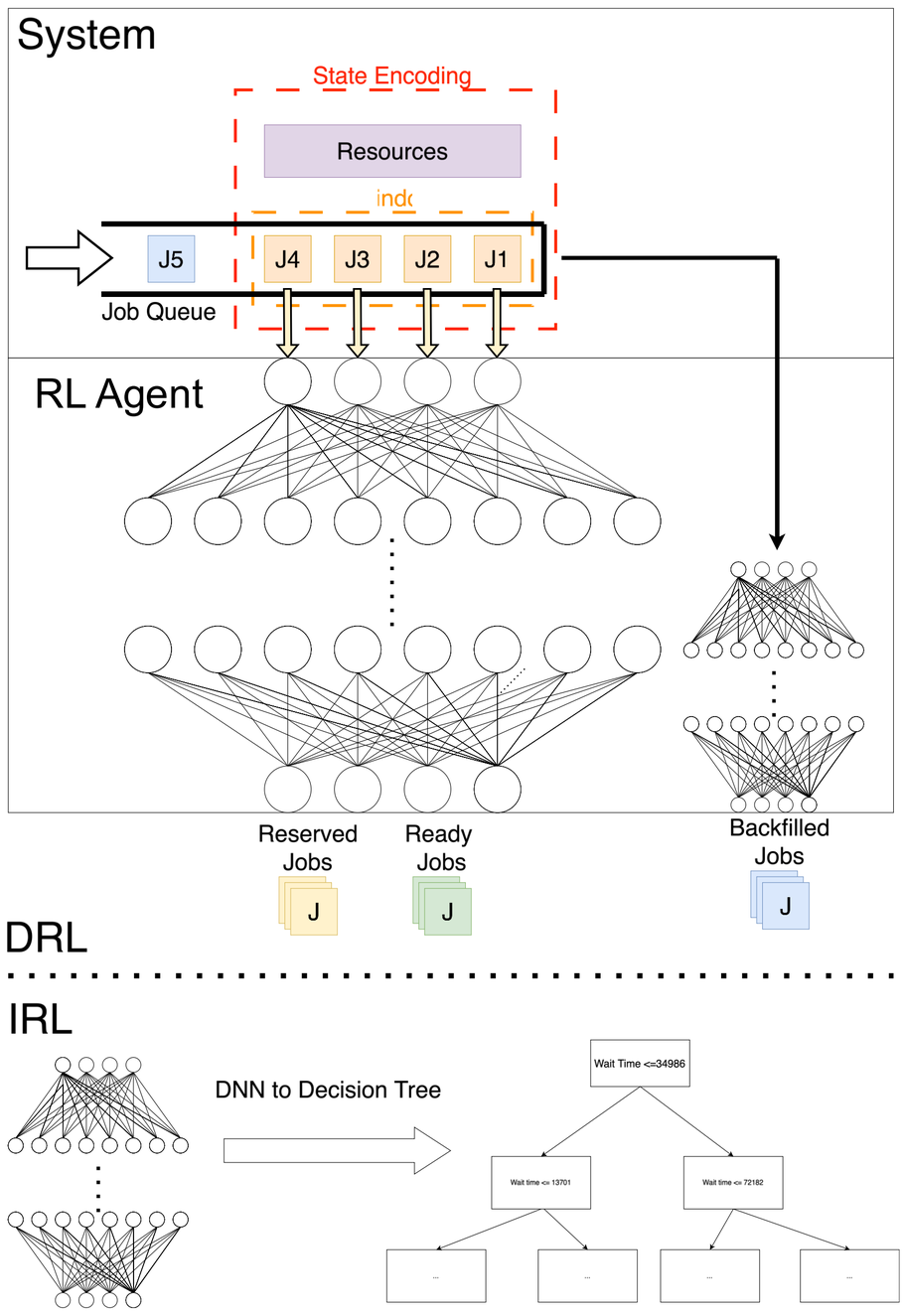
- Utilizes a hierarchical neural network to handle job scheduling and backfilling in HPC systems
- Incorporates window-based scheduling and EASY backfilling to enhance system utilization and avoid job starvation.
- Demonstrated significant performance improvement over traditional heuristics.

Y. Fan, Z. Lan, T. Childers, P. Rich, W. Allcock and M. E. Papka, "Deep Reinforcement Agent for Scheduling in HPC," *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Portland, OR, USA, 2021, pp. 807-816

Interpretable RL

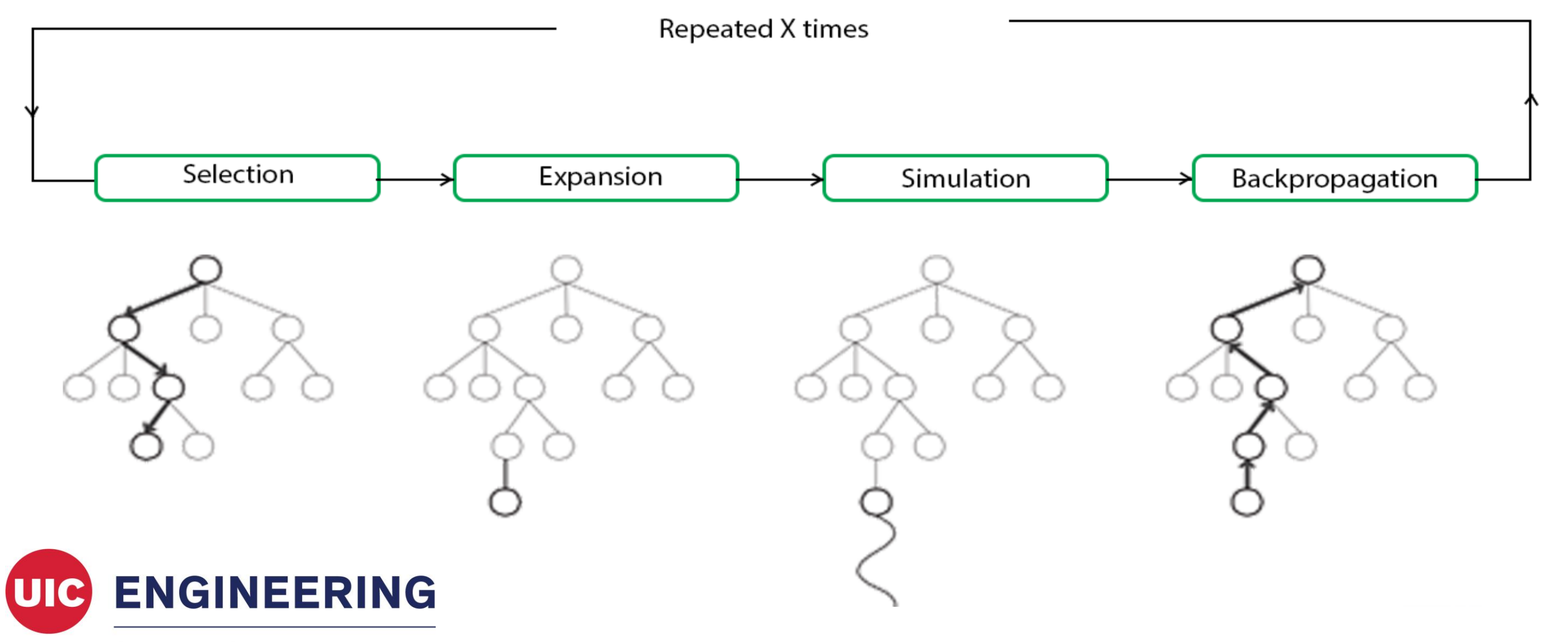
- Converts DRL scheduling policies into interpretable decision trees using imitation learning.
- Employs the DAgger algorithm and the concept of "critical states" to reduce tree size while maintaining performance.
- Provides transparency and aids system administrators in understanding and validating scheduling policies.

B. Li, Z. Lan and M. E. Papka, "Interpretable Modeling of Deep Reinforcement Learning Driven Scheduling," in 2023 31st International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Stony Brook, NY, USA, 2023, pp. 1-8



Monte Carlo Tree Search (MCTS)

- MCTS is a heuristic search algorithm that builds a search tree based on simulations.
- Easy to interpret and customizable by system administrators.
- Unlike DRL, which requires intensive training and generalization over large state spaces, MCTS leverages a known simulator for direct.



MCTS Stages

- **Selection**
  - The goal is to balance **exploration** and **exploitation**.
- **Expansion**
  - From the selected node, create one or more **child nodes** by exploring **unvisited actions**.
- **Simulation**
  - Simulate a random or **semi-random playout** (sequence of actions) until reaching a terminal state.
- **Backpropogation**
  - The result of the simulation (reward) is propogated up the tree.

This repeats for several iterations.

UIC ENGINEERING
Computer Science