Lore: A Learning-based Approach for Workflow Scheduling in Clouds

Haosong Peng, Chuge Wu, Yufeng Zhan, and Yuanqing Xia

Haosong peng
`livion_i@icloud.com`

School of Automation
Beijing Institute of Technology

September 11, 2022

Introduction

BEIJING INSTITUTE OF TECHNOLOGY

Many computing jobs such as deep learning model inference can be regarded as workflows. They encode job stages and their dependencies as directed acyclic graphs (DAGs)

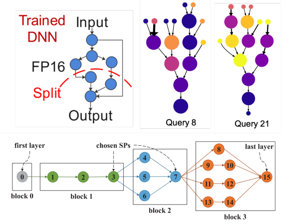


Figure: Workflows in deep learning

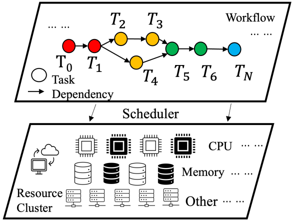


Figure: Resource cluster environment

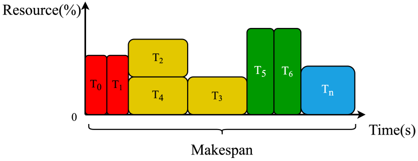


Figure: Example of workflow execution process

Objective: minimize the completion time (*makespan*) by resolving the best placement order of tasks.

Problem Formulation

Given a workflow that includes a set of tasks $\mathcal{T} = \{T_1, T_2, \ldots, T_N\}$.

- configuration tuple of task T_i: $(t_i, cpu_i, mem_i, exe_i)$
- parent tasks of T_i: $\mathcal{T}_i^{pred}$
- resource amount of cluster: CPU_{res} and Mem_{res}.

$$
\begin{aligned}
\min \quad & makespan, && (1)\\
s.t. \quad & \sum_{T_i \in \mathcal{T}_e} cpu_i \le CPU_{res}, && (2)\\
& \sum_{T_i \in \mathcal{T}_e} mem_i \le Mem_{res}, && (3)\\
& exe_j = 1, \forall T_j \in \mathcal{T}_i^{pred}, \forall T_i \in \mathcal{T}_e. && (4)
\end{aligned}
$$

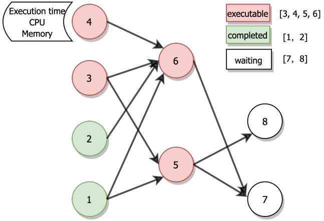


Figure: DAG example

Eq. (2) and (3) represent the resource constraints, and Eq. (4) represents the dependency constraints of workflow.

Formulation as MDP

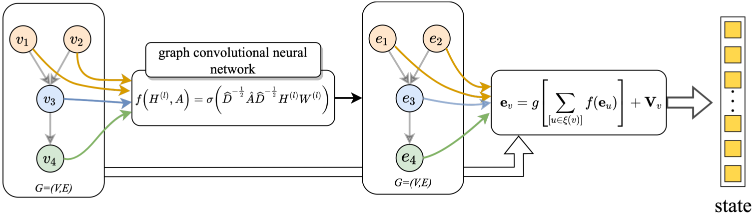


Figure: Graph feature extraction.

- **State.**
  1. current time t
  2. remaining CPU resources: CPU_res
  3. remaining memory resources: Mem_res
  4. tuples of *ready task (executable tasks list of length M)*: T_a
  5. graph features extracted from graph neural network

Formulation as MDP

BEIJING INSTITUTE OF TECHNOLOGY

- **Action.**

$$\{-1, 1, 2, 3, \dots, M\}$$

1. $a = -1$: execute the tasks of the shortest time consumption in the cluster.
2. $a = 1 \sim M$: schedule the a-th task in the current *ready task* list to the cluster.

Then update the *ready task* list.

- **Reward.**

1. $a = -1$: $reward = -$task T_i' s makespan
2. $a = 1 \sim M$: $reward = 0$

Lore's DRL framework

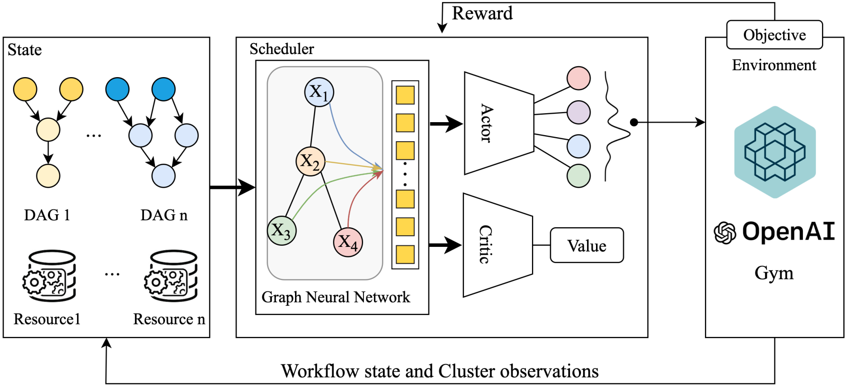


Figure: Lore framework.

Monte Carlo Tree Search with DRL

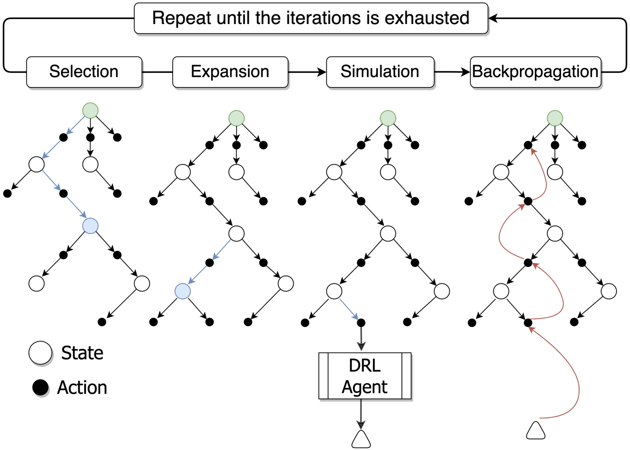
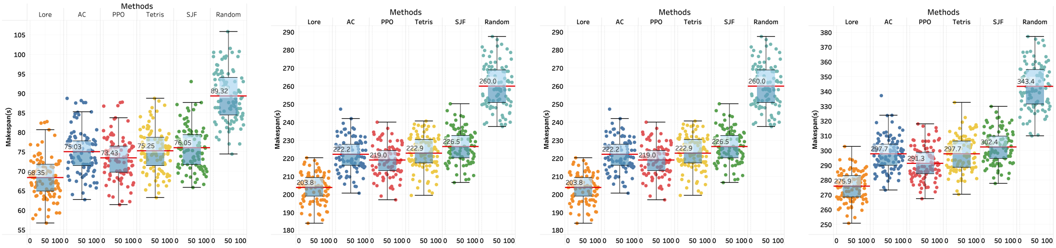


Figure: An improved MCTS algorithm with DRL agent

Makespan performance

BEIJING INSTITUTE OF TECHNOLOGY

In each round, we generate 1000 DAGs, and each DAG has N tasks ($N = 10, 20, 30, 40, 50$) with various duration and heterogeneous resource demands for CPU and memory. We test the makespan of Lore and the other 5 baseline approaches.



(a) $n = 10$ (b) $n = 20$ (c) $n = 30$ (d) $n = 40$

Figure: Performance of makespan in different DAG sizes (the lower value, the better performance.)

Performance on makespan shows the basic conclusion:

$$Lore > PPO > AC \approx Tetris > SJF > Random$$

Conclusion

In this paper, we propose a DRL-based approach to solve the problem of minimizing the makespan of cloud workflows.

1. We establish the system model of workflow scheduling and transform it into an optimization problem.
2. A new DRL-based approach with MCTS is designed to solve the problem.
3. The results show that Lore outperforms other baseline strategies, resulting in a $2 - 10\%$ reduction in makespan and high resource utilization of up to 20% in different cases.

Thank you for your attention!