Learning Scheduling Algorithms for Data Processing Clusters

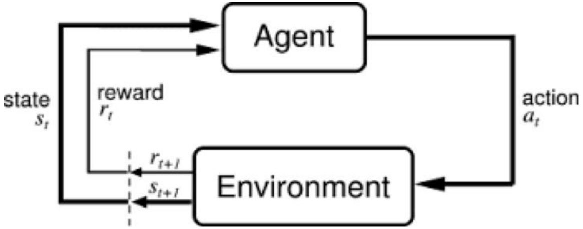
Aakhila Shaheen

Introduction

- Cluster schedulers prioritize generality, ease of understanding over achieving ideal performance
- Efficient utilization of compute resources can save millions of dollars at scale
- Schedulers today are oblivious to the underlying problem statement when designing scheduling policies
- The authors propose Decima, a general-purpose scheduling service for data processing jobs with depend stages using Deep Reinforcement Learning and Neural Networks

Reinforcement Learning

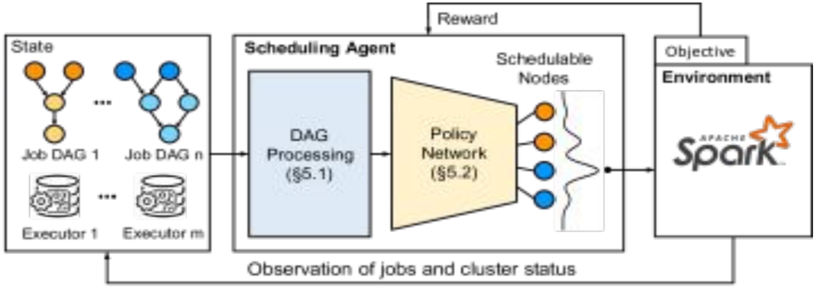
- An area of machine learning which is concerned with how agents ought to take actions in an environment so as to maximise some notion of cumulative reward



- The goal of reinforcement learning is to pick the best known action for any given state.
- Statistically, its an attempt to model a complex probability distribution of rewards in relation to a very large number of state-action pairs

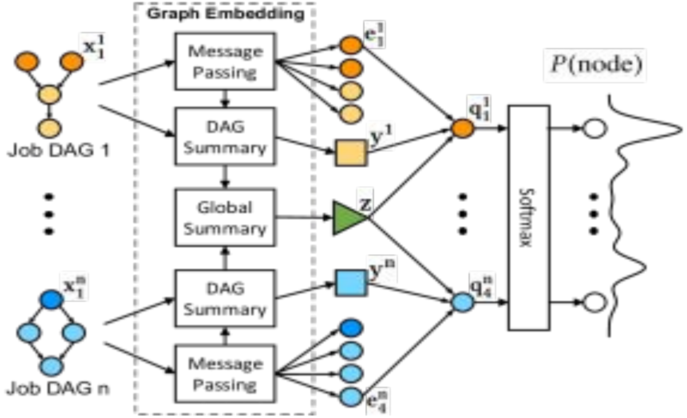
Decima- The Big Picture

- Given only a high level objective(e.g, minimal average job completion time) Decima uses existing cluster monitoring information and past workload logs to automatically learn sophisticated policies
- It learns to use jobs dependency structure to plan ahead and avoid waiting at choke points
- It also learns job level parallelism to avoid wasting resources on diminishing returns for jobs with little inherent parallelism



Processing DAG Inputs

- Decima uses a new embedding technique for mapping job DAGs with arbitrary size and shape to vectors that neural networks can process
- It is built on recent work on learning graph embeddings but is tailored to scheduling domain



Processing DAG inputs

The graph embedding outputs three different types of embeddings:

- Per-node embedding : Capture graph structure by embedding information about node and its children
- Per-job embedding : Aggregate information across the entire job
- Global embedding: Combines information from all job-level summary into cluster level summary

Importantly, the information to be stored in these embeddings is not hardcoded but Decima learns it from its input DAG's through end-to-end training.

Encoding Scheduling Decisions

- Need to balance between the naive “executor-centric approach” and the more complex joint probability distribution of partitioned executors and the available jobs in the system

Decima decomposes scheduling decisions into a series of two-dimensional actions which output

- A stage designated to be scheduled next
- A cap on the maximum allowed parallelism for that stage’s job.

Handling Continuous Stochastic Job Arrivals

Training Decima for continuous job arrivals creates 2 challenges:

- Standard RL objective of maximizing the expected sum of rewards is not a good fit

  Use an alternative RL formulation that optimizes for the *average reward* in problems with an infinite time horizon

- Different job patterns have a large impact on reward feedback

  Account for the variance caused by the arrival processes by building upon recently-proposed variance reduction techniques for “input-driven” environments [variance-reduction]

Interesting Finds

The key contributions of this paper are:

- Novel scalable graph processing techniques that convert job DAGs of arbitrary shape and sizes into vectors feasible for neural network and end-to-end RL
- Introduce variance reduction techniques to make RL training feasible for unbounded job arrival sequences
- It is the first generalisable, RL based scheduler that schedules complex data processing jobs without human-encoded inputs