Resource Management with Deep Reinforcement Learning

Hongzi Mao, Mohammad Alizadeh, Ishai Menache, Srikanth Kandula

Yashovardhan Sharma
Problem
Problem...

• Resource management problems in systems

• Current solutions are complicated and take painstaking effort to implement

• Can systems learn to manage resources on their own?
Solution
What do they suggest?

• Use Machine Learning (obviously!)

• But do we need ML for this problem?

• They argue:

  1. Underlying systems are complex; hard to model accurately

  2. Have to make online decisions with noisy inputs; should work well under diverse conditions

  3. Some performance metrics are hard to optimise in a principled manner
Key Idea

• Reinforcement Learning

• Agent learns directly from experience, by interacting with the environment

• They believe that this approach is particularly well-suited to resource management systems
  • Why?
Key idea...

- Create **DeepRM** - a simple multi-resource cluster scheduler
- It learns to optimise various objectives. e.g. minimise avg. job slowdown, completion time
- Doesn’t require any prior knowledge of system behaviour to learn these strategies
- Can support a variety of objectives simply by using different reinforcement awards
RL 101

Agent

DNN

policy
\(\pi_\theta(s, a)\)

Environment

Reward \(r\)

Take action \(a\)

Observe state \(s\)

Parameter \(\theta\)
DeepRM

- Learns by performing gradient-descent on the policy parameters

- Objective is to maximise the cumulative discounted reward

- Gradient of this objective is given by:

$$
\nabla_{\theta} \mathbb{E}_{\pi_{\theta}} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right] = \mathbb{E}_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(s,a)Q^{\pi_{\theta}}(s,a)]
$$
Problem Formulation

Resources:
- CPU
- Memory

Time:

Clusters:
- Green
- Red
- Purple

Jobs:
- Slot 1
- Slot 2
- Slot 3
Results
Summary

• DeepRM is comparable to and often better than all heuristics

• DeepRM is the best performing scheme on each objective when trained specifically to optimise for that objective with the appropriate reward function

• Feasible to apply state-of-the-art Deep RL techniques to large-scale systems
Critique

- No preemption and no malleability
- Preference given to early-arriving jobs
- Resource profile of jobs may not be known in advance
Questions?