Resource Management with Deep Reinforcement Learning

Hongzi Mao, Mohammad Alizadeh, Ishai Menache, Srikanth Kandula

Yashovardhan Sharma

UNIVERSITY OF CAMBRIDGE

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

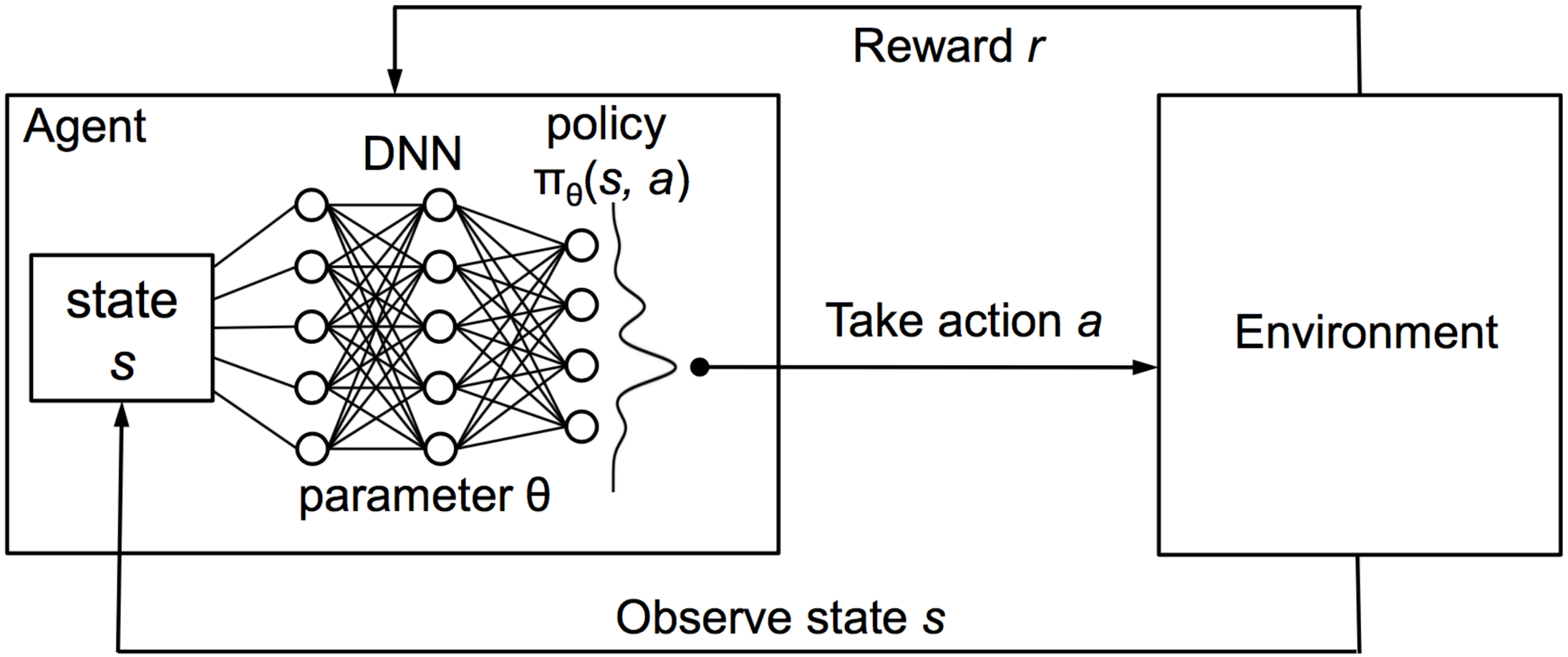
Key Idea

- **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

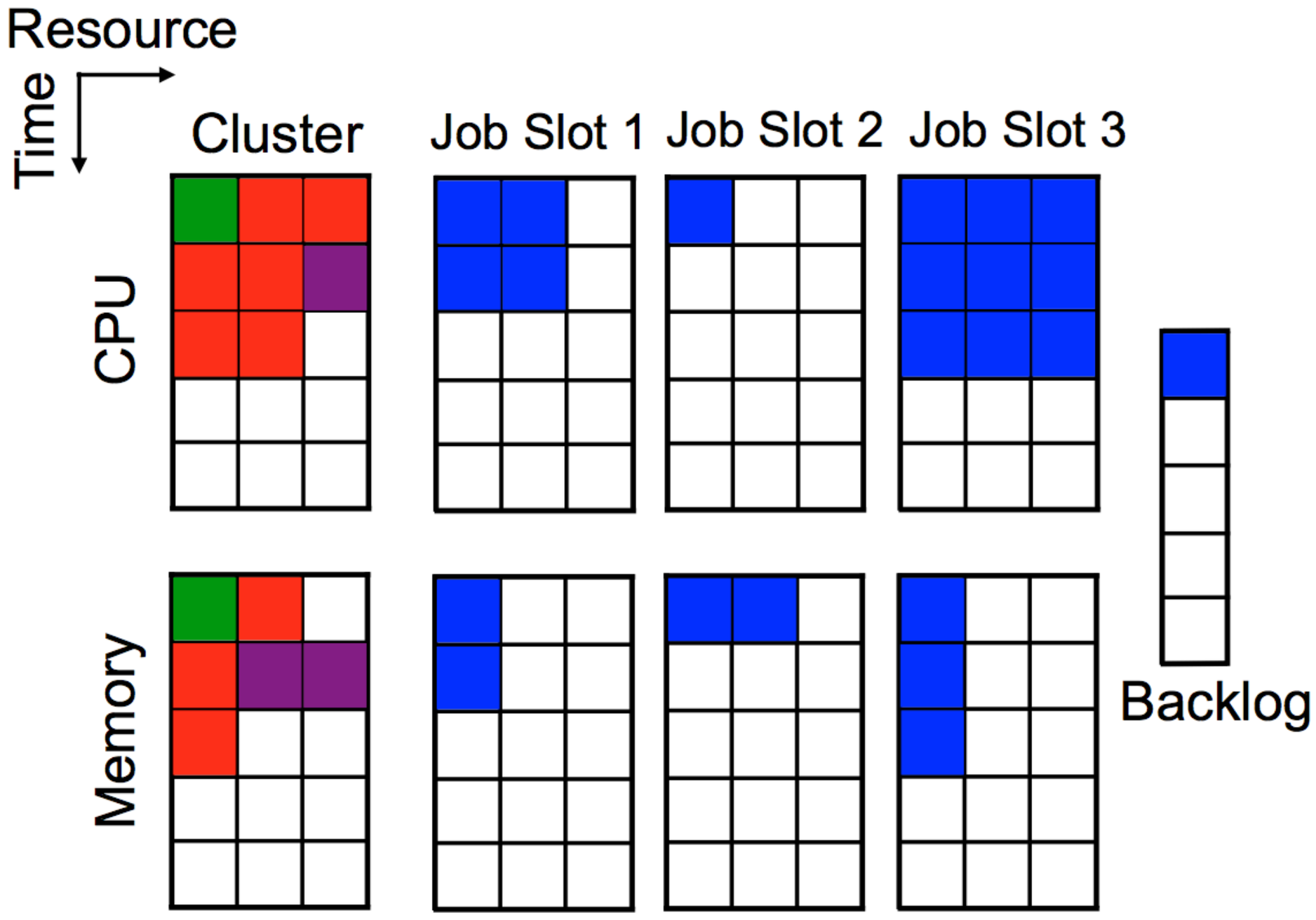

Reward r
Agent
DNN
policy
$\pi_\theta(s, a)$
state
s
parameter θ
Take action a
Environment
Observe state s

DeepRM

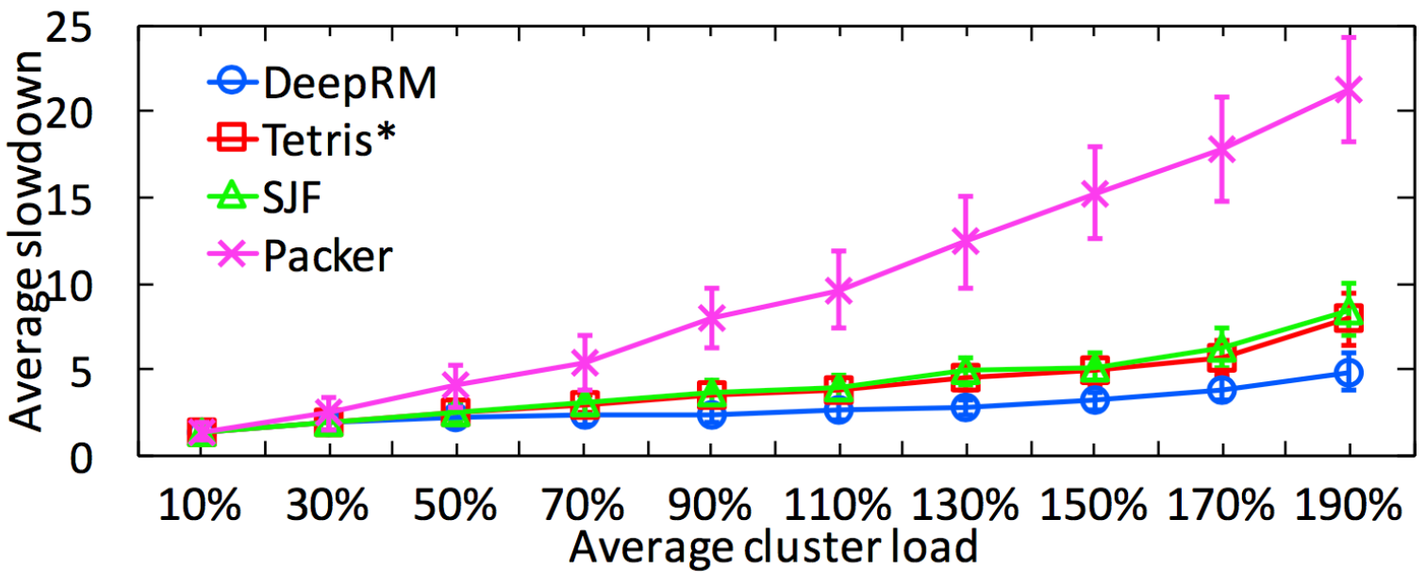
- Learns by performing gradient-descent on the policy parameters
- Objective is to maximise the cumulative discounted reward
- Gradient of this objective is given by :

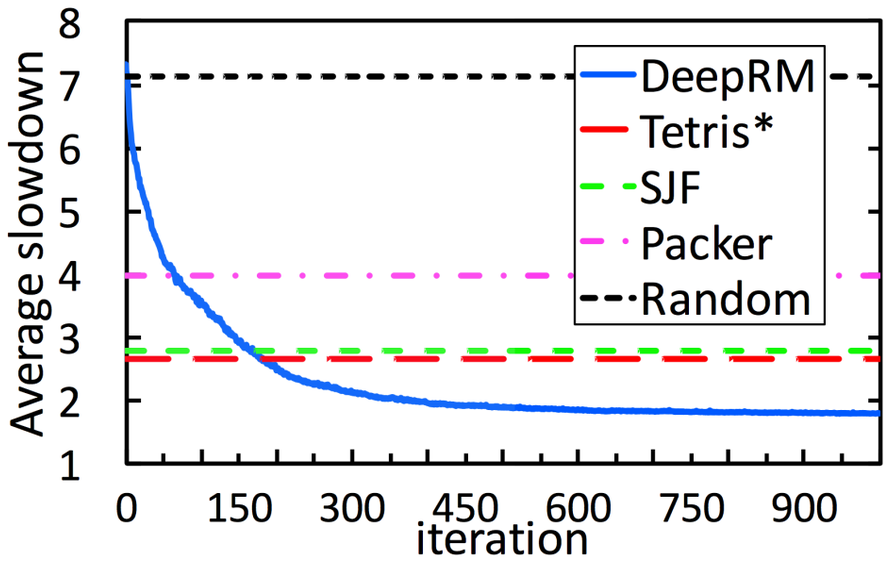
$$\nabla_\theta \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] = \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \log \pi_\theta(s,a) Q^{\pi_\theta}(s,a) \right]$$

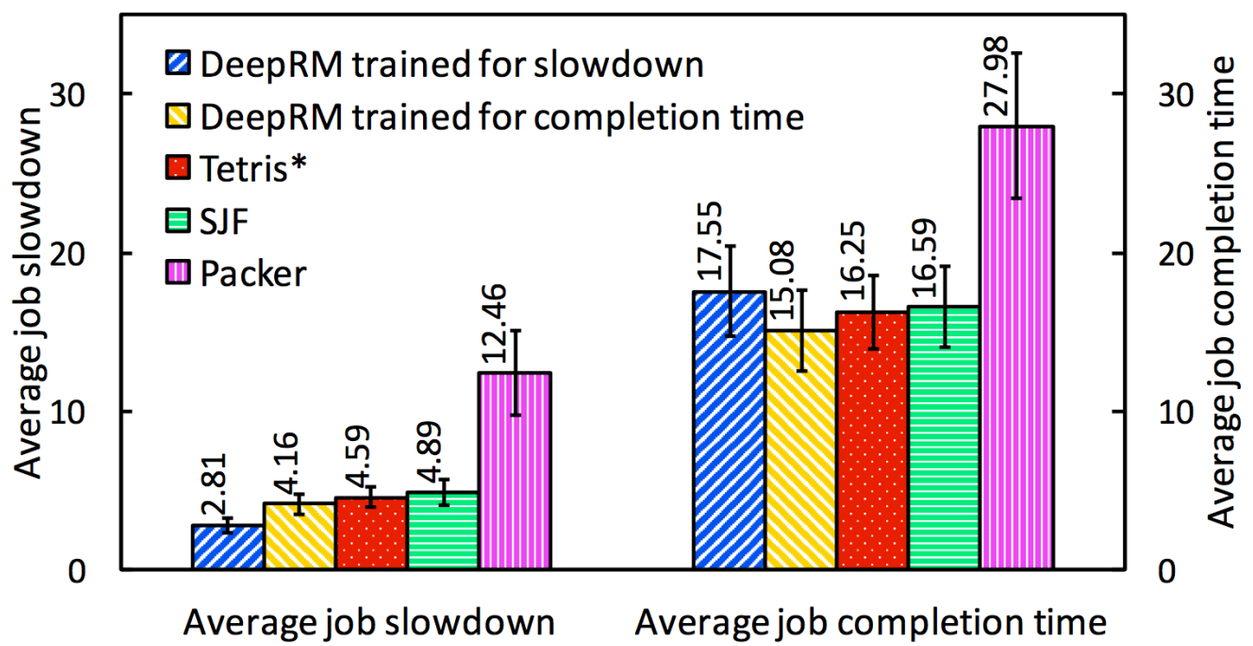
Problem Formulation



Results


DeepRM
Tetris*
SJF
Packer
Average slowdown
0
5
10
15
20
25
10%
30%
50%
70%
90%
110%
130%
150%
170%
190%
Average cluster load


DeepRM
Tetris*
SJF
Packer
Random
Average slowdown
1
2
3
4
5
6
7
8
0
150
300
450
600
750
900
iteration


DeepRM trained for slowdown
DeepRM trained for completion time
Tetris*
SJF
Packer
Average job slowdown
Average job completion time
0
10
20
30
2.81
4.16
4.59
4.89
12.46
17.55
15.08
16.25
16.59
27.98
Average job slowdown
Average job completion time

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?