GreenDRL: Managing Green Datacenters Using Deep Reinforcement Learning

Kuo Zhang, *Peijian Wang, Ning Gu, Thu D. Nguyen*
Rutgers University

RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

Background

- Datacenters (DCs) account for 1% of worldwide electricity use in 2018, and the demand for DCs keeps increasing

Operational cost

Carbon emission

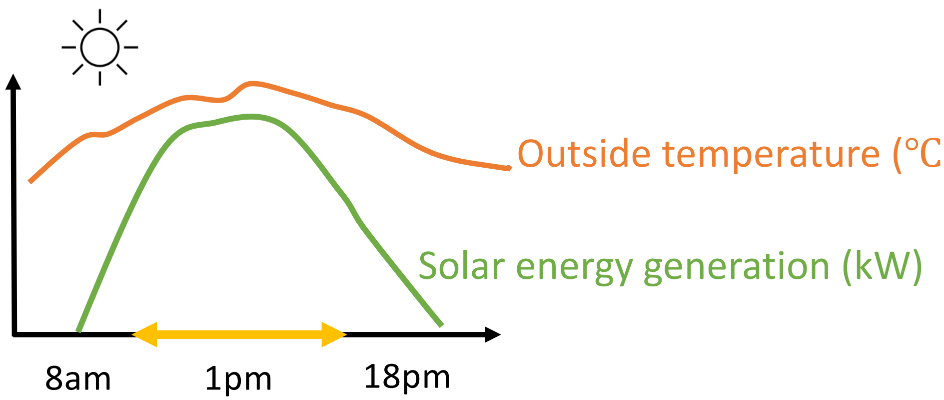
- **Green DCs** is one way to increase the sustainability of DCs

Motivation

Managing green DCs to maximize the benefit is complex and challenging:

Minimize brown energy	**Use green energy efficiently**
Meet workload SLA	**Efficient cooling control**

Multiple aspects, and complex tradeoff space



More workload execution to use green usage ☺
More cooling energy need ☹

We propose to use deep reinforcement learning technique to jointly manage several important aspects of the DC operation

Our Solution: GreenDRL

- A deep reinforcement learning (RL) based management system that **jointly** manages
  - server energy
  - cooling control
  - workload scheduling
- Can be systematically optimized for performance via training
  - Thus, reduces the effort to design handcrafted heuristics
- Does not depend on predictions of the future

Problem Description

Consider a green DC:

- **Two types of power source**:
  - on-site green energy (free of cost) generation
  - brown energy from power grid
- **Hybrid cooling system**:
  - free cooling unit
  - Air Conditioner
- **Compute-intensive workload consists of two types of jobs:**
  - deferrable job: can be delayed by *x* hours
  - nondeferrable job: should execute as soon as possible
- **Servers:** can be put into inactive low power state (e.g., ACPI S3)

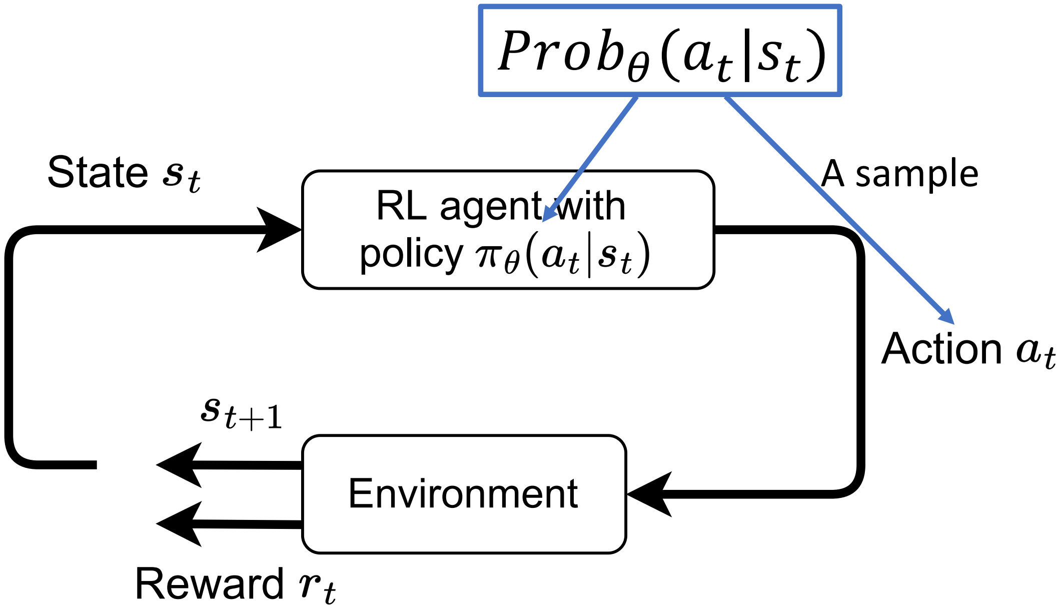
Objective

1. Maintaining internal temperature
2. Minimize waiting times for nondeferrable jobs
3. Minimize delaying deferrable jobs for longer than the threshold delay time period
4. Minimize brown energy cost

More important

Less important

RL Background and Design Considerations



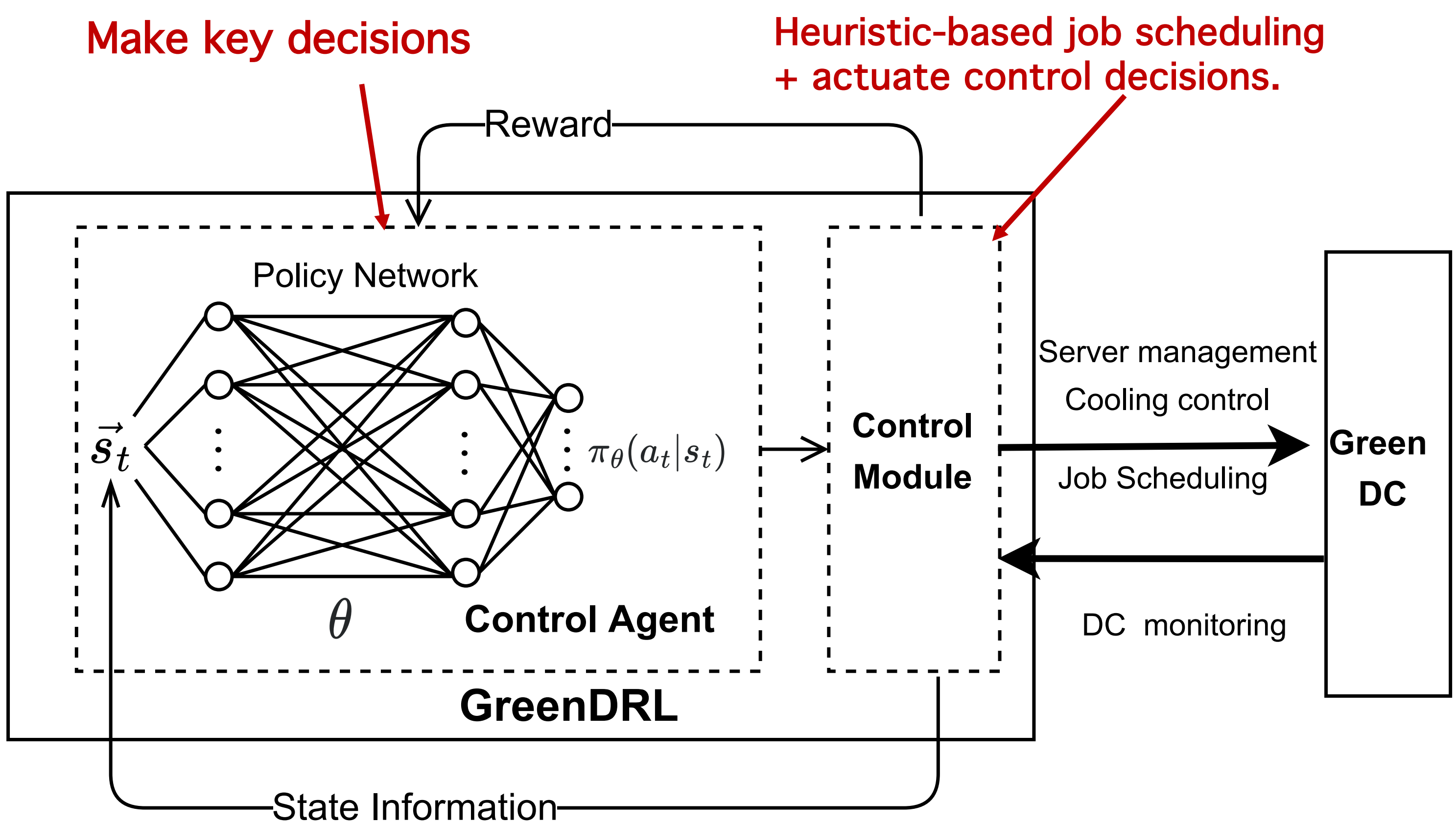
RL

The goal: learn a policy , $\pi_\theta(s_t, a_t)$, that maximizes the expected cumulative reward

Many control decisions and extremely large learning space

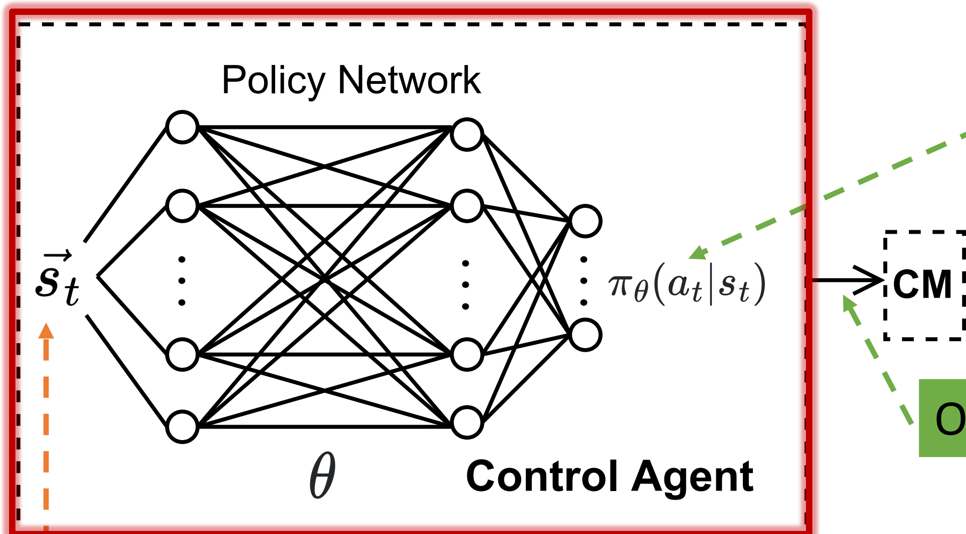
Scalable and adaptable as the DC builds out or change

GreenDRL Design: Two Components Partition



GreenDRL Overview

GreenDRL: Control Agent (CA)



$\pi_\theta(a_t|s_t) = \Pr_\theta(a_t|s_t)$:

$$\begin{pmatrix} \text{Cooling} \\ \text{Server allocation} \\ \text{Deferrable Job dispatch cap} \end{pmatrix} \sim \mathcal{N}ormal_\theta$$

Output a sample, a_t

CA's neural network is relatively structurally independent of the problem size, e.g., # of servers, # of cooling operations.

- Green energy in previous time slot
- Current in and outside temperatures
- Cluster load
- …

$$Reward_t = f_{linear}(Penalties_t, EnergyCost_t)$$

GreenDRL: Control Module(CM)

- Mapping actions and actuate DC operations, e.g.,

"Set free cooling at 50% fan speed"

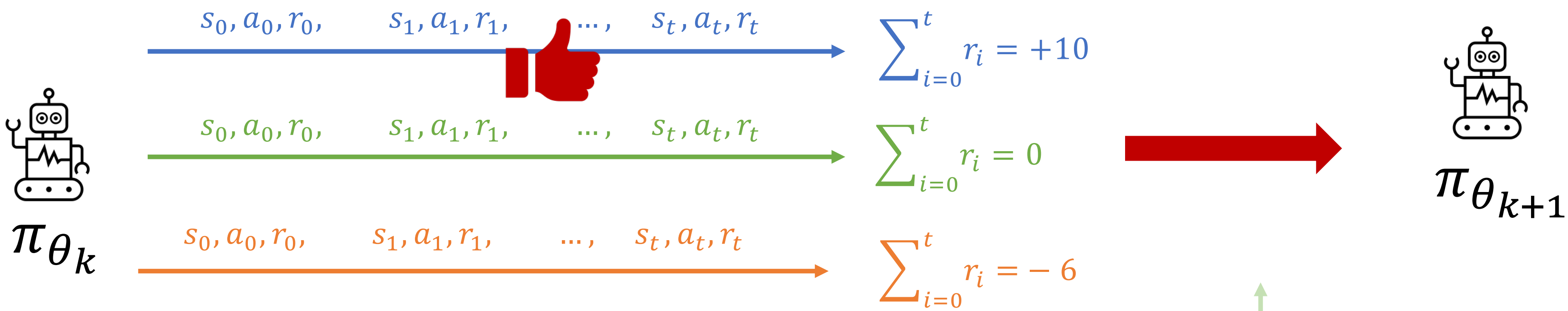
"keep N server active"

- Heuristic-based job scheduling:
  - Respect CA's decision
  - Prioritize nondeferrable jobs
  - Packing active servers to increase utilization

The workings of CM is part of CA's environment:

- any well-known job scheduling heuristic can be used

GreenDRL Training: The Intuition



Interact with a DC simulator to generate ***trajectories*** with current policy π_{θ_k}

Increase the probability of "Good" trajectory by updating θ_k in the neural network (gradient ascent update)

More customized implementation to make training stable and effective in the paper

Evaluation: Build a Green DC Simulator

- Build a simulator for *Parasol,* including:
  - Server power model
  - Cooling thermal models for both free cooling and AC

Parasol testbed

Challenges: During training, the RL may explore abnormal situations that are never seen in a normal DC, e.g.,

- Turn on AC when inside temperature is just 5℃

We study physical thermal theory.
Define models with reasonable behaviors even in less-seen situation

Evaluation

Overall evaluation setup	
Workload trace	Google and Alibaba
Environmental trace	Parasol traces with different weather patterns
Servers #	32
Time slot duration	5 min
Deferrable load vs nondeferrable	75% v.s. 25%
Deferrable deadline	12 hours

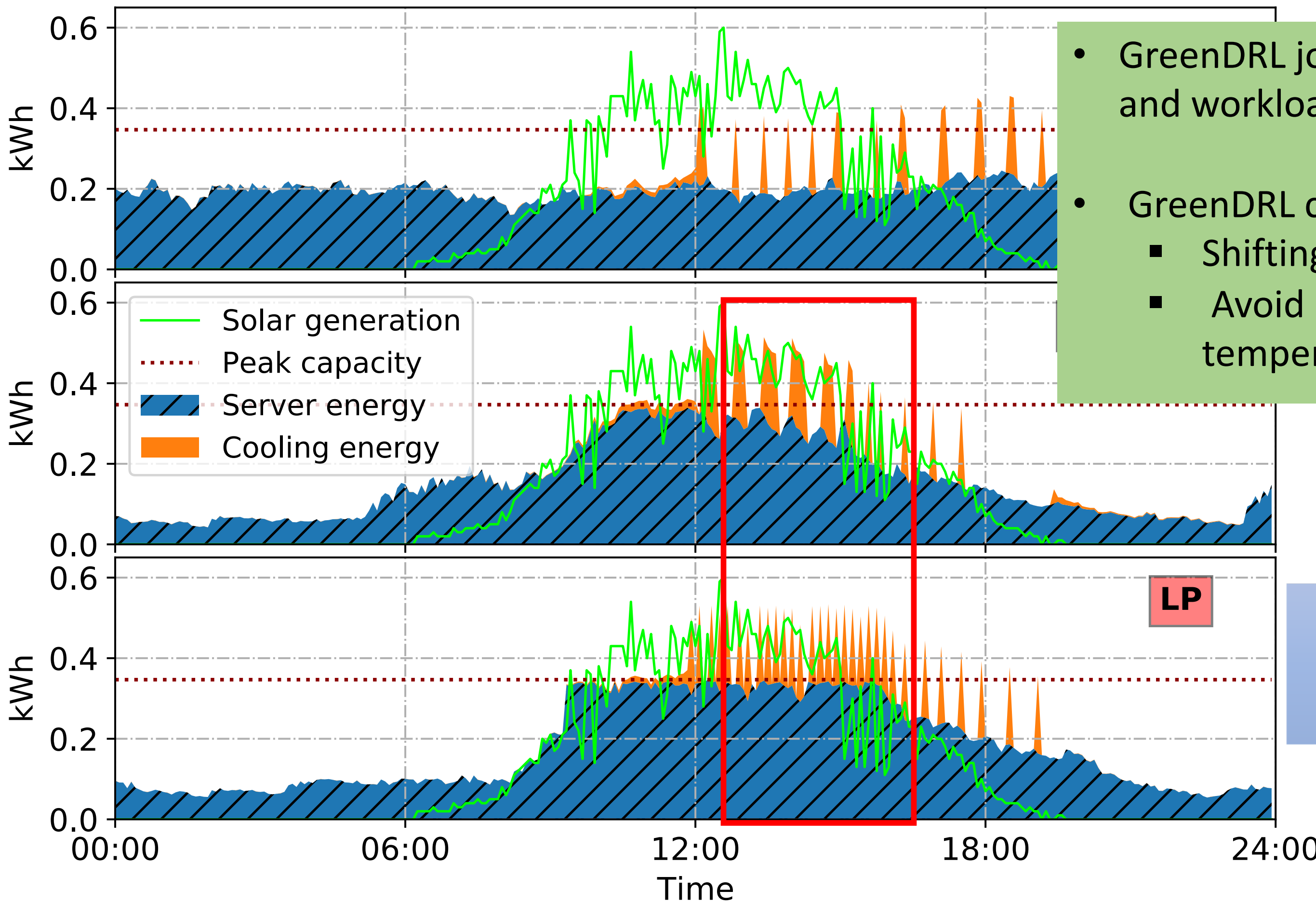
Baselines:

FIFO: Always keep just-enough servers to execute the workload + Simulation of a commercial cooling controller

LP: MILP with perfect future knowledge (adapts *GreenSwitch* [ASPLOS'13]) + adapted CoolAir[ASPLOS'15] for cooling control.

Results

A HighSolar-HighTemperature day



- GreenDRL jointly managing cooling, server power and workload leads to benefit
- GreenDRL can learn basic principles:
  - Shifting deferrable workload
  - Avoid missing deadlines, maintain temperatures efficiently (results in paper)

Energy Saving:
- *54% compared to FIFO*
- *24% compared to LP*

More Results in the Paper

- Can learn coordinated cooling and server allocation decisions
- 18% energy saving over a year compared to FIFO

And More:

- Scalability
- Impact of different weights in the reward function
- Sensitivity to
    - cluster load
    - defer vs nondeferrable ratio
    - deferrable workload delay tolerance

Conclusion

- We study the use of deep RL to jointly manage several important controllable aspects of a green DC operation
- GreenDRL combines a deep RL agent and simple heuristics
- Simulation results using historical data collected from Parasol, an experimental green DC shows:
  - GreenDRL can successfully learns important management principles
  - Outperforms two baseline policies

Thank you!