SibylOpt: Managing Green Data Centers Using Off-Online Deep Reinforcement Learning

Ning Gu[1] Kuo Zhang[1] Thu D. Nguyen[1] Peijian Wang[1] Tania Lorido-Botran[2,3]

[1]Rutgers University [2]Roblox [3]Northeastern University

Motivation

- Data centers (DCs) are significant electricity consumers → high operational costs and carbon emission
- Green DCs incorporate features such as renewable energy and efficient cooling for increased sustainability
- Managing green DCs is challenging
- Deep reinforcement learning (DRL) offers robust approach to optimize management policy for specific workload and DC characteristics
- How to train management agent?

SibylOpt: Off-Online DRL

- Uses **offline RL** to train using historical traces collected using any policy (e.g., heuristic-based policies)
- Applies **online RL** post-deployment to improve/fine-tune and adapt to evolving workload and environment
- Continuously expands training dataset during operation, enabling progressive adaptation to system changes

Design Overview

- **Objectives:** Optimize job scheduling and cooling in a green DC to
  - Maintain inlet temperature $\leq 30°C$
  - Minimize delay for nondeferrable jobs
  - Prevent starvation of deferrable jobs
  - Minimize use of grid electricity
- **Architecture:**
  - Control Agent (CA): DRL agent making control decisions based on current state of DC
    - Inputs: Temperature, solar power, job queues, time, date, etc.
    - Outputs: Number of active servers and cooling system settings
  - Control Module (CM): Applies actions, dispatches jobs, and handles thermal safety
    - Manages individual server power states (on/suspended)
    - Handles job dispatch to active servers
    - Takes emergency actions during thermal events
- **Reward:** Balances temperature, job delay, and grid energy consumption

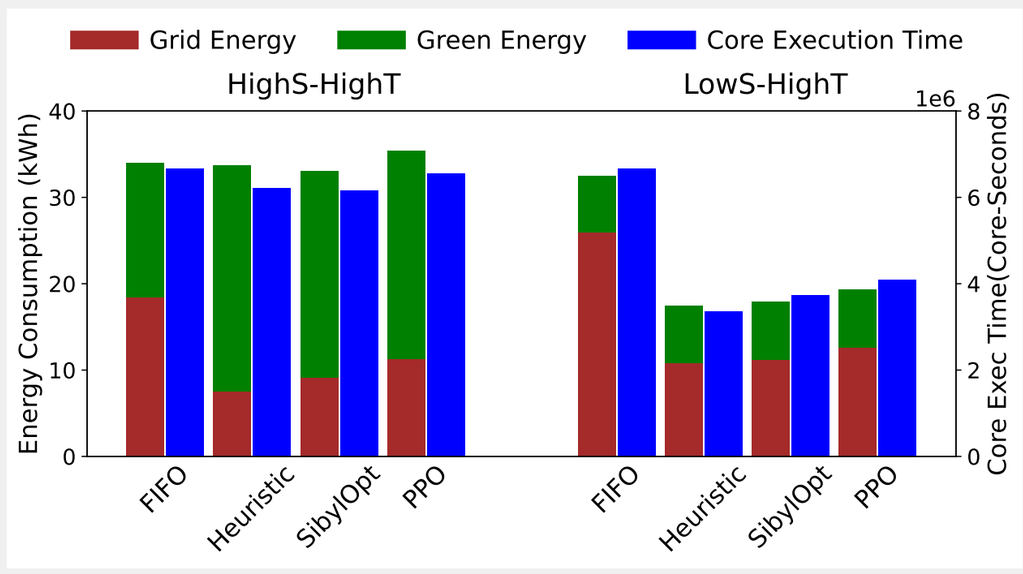
AWAC: Offline RL Algorithm

- **Purpose:** Train DRL policies offline without requiring real-time environment interactions
- **Overview:**
  - Learns from static dataset $D = (s, a, r, s')$
  - Leverages advantages $A(s, a) = Q(s, a) - V(s)$ for stable updates
  - Caps advantage with A_{max} to avoid large policy shifts
- **Components:**
  - **Actor:** Proposes actions given states
  - **Critic:** Evaluates $Q(s, a)$ to guide learning
  - **Target Critic:** Stabilizes updates via delayed synchronization
- **Training:** Conducted entirely offline, with optional online phase for post-deployment adaptation

Evaluation

- **Simulator:** Models power/cooling dynamics using trace data from Parasol, a solar-powered micro-DC with hybrid cooling system
- **Baselines Policies:**
  - **FIFO:** Activates servers on demand, suspends after idle threshold; threshold-based cooling
  - **Heuristic:** Solar-aware server activation proportional to available solar, job deferral
  - **PPO:** Based on GreenDRL, a previous DRL approach requiring online RL (need simulator for training)
- **Training Data:**
  - Weather: 270-day trace collected by Rutgers Weather Station
  - Workload: Scaled Google cluster trace (26% average utilization, 75% deferrable jobs)
  - Training data collected using Heuristic policy
- **Evaluation Scenarios:**
  - 4-day traces under varied solar and temperature conditions not included in training data
  - Full-year performance evaluation

Evaluation Results



Core-hours and energy use (solar/grid) for HighS–HighT and LowS–HighT scenarios

Key Findings

- All policies except FIFO attempt to defer jobs to solar-rich periods
- **SibylOpt** consistently reduces grid electricity usage and improves efficiency compared to FIFO
- **SibylOpt** achieves higher rewards than its behavior policy (Heuristic) in most scenarios
- **SibylOpt** and PPO learn to jointly manage cooling and server power to maximize solar energy utilization
- **Performance patterns:**
  - On solar-rich days: Effective job deferral to match solar availability
  - On cloudy days: Strategic job deferral to future days while avoiding starvation
- **Importance of behavior policy:**
  - Offline learning leads to poor performance if desirable states (e.g., job deferral under solar) are absent
  - E.g., Random and FIFO behavior policies result in poor offline performance
- **SibylOpt** offers competitive performance with PPO without requiring a detailed simulator for training