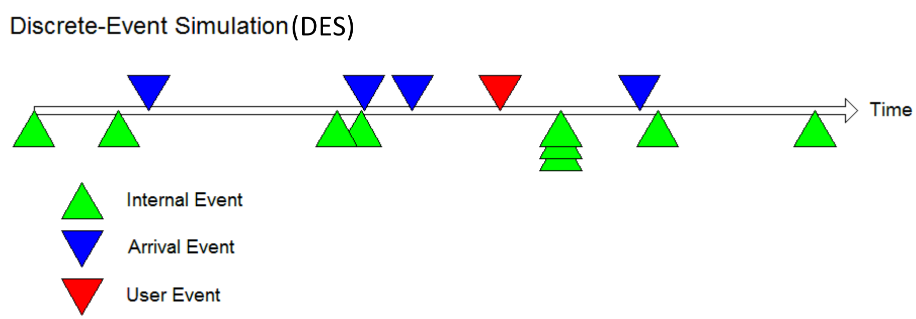
Intelligent Resource Management for Energy Efficient Computing

Zhiling Lan (UIC & ANL)

Intro

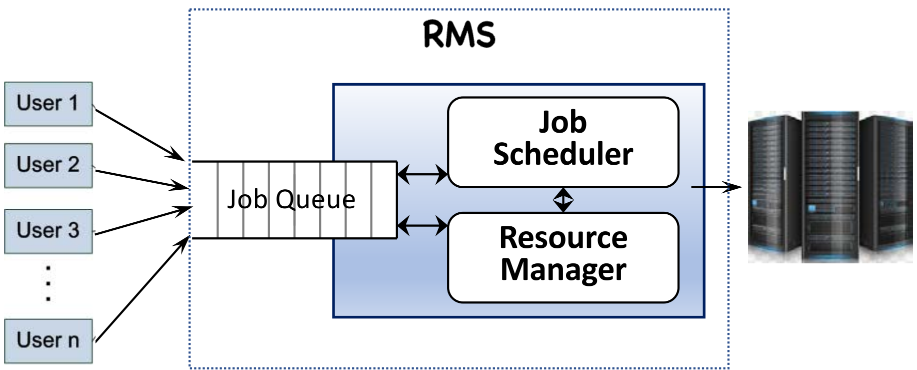
- Resource management and scheduling
  - In partnership with the Cobalt (now PBSPro) team at ALCF
- Modeling and simulation
  - Cluster scheduling
  - Interconnect networking



A DES based cluster scheduling simulator: https://github.com/SPEAR-UIC/CQSim

Resource Management and Scheduling (RMS)

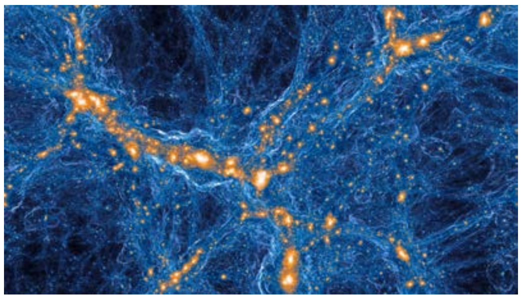
- Different terms: workload management, batch scheduling, cluster management
- The same design since 1990s
  - For **traditional numerical simulations** on **homogeneous systems**
  - **Bare metal** mode with **exclusive node** access
  - **FCFS, along with resource reservation and backfilling**
  - Common metrics: node utilization, job wait time, …



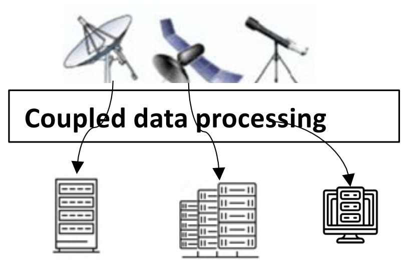
In cloud computing, users are given time-controlled ownership of resource containers via VM.

Workload Evolution

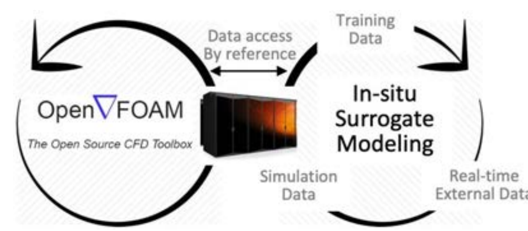
Traditional numerical simulation



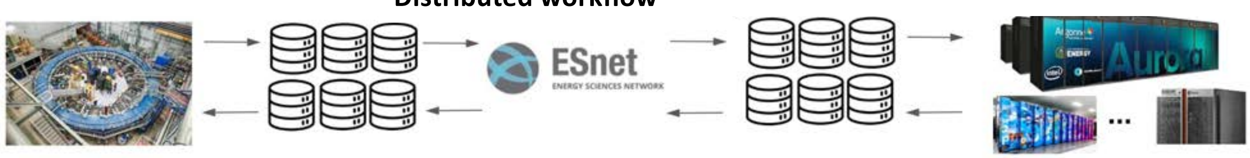
AI/ML/Data Analytics



AI-enabled Science

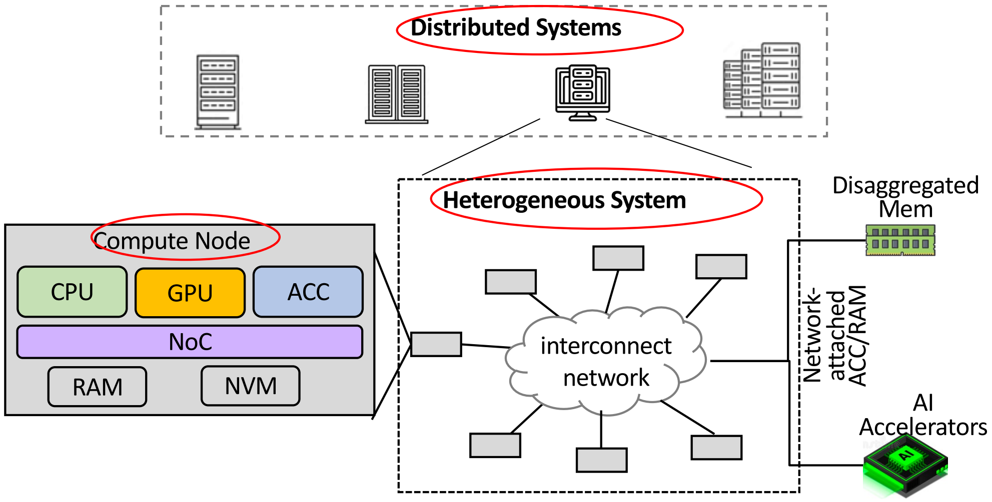


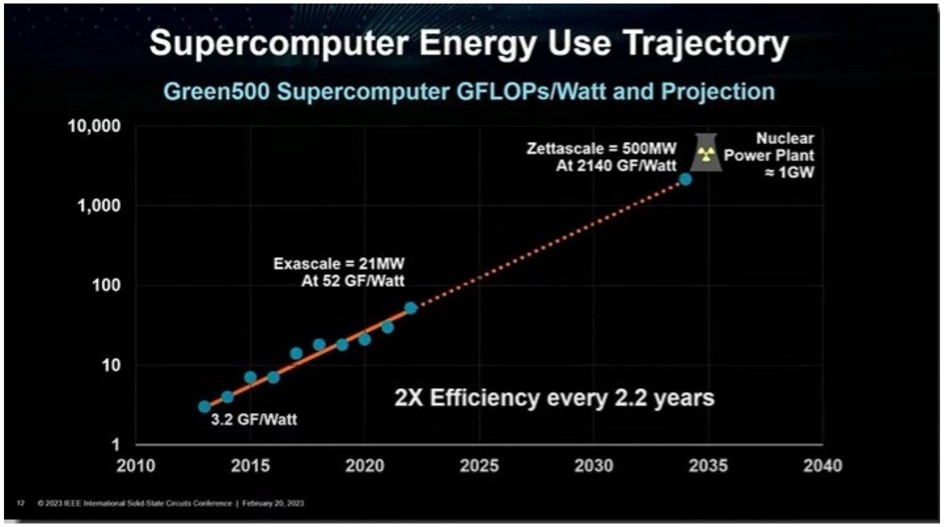
Distributed workflow



Hardware Evolution

Heterogeneity is manifested across all levels, from chip to multi-system, and in each component per level.

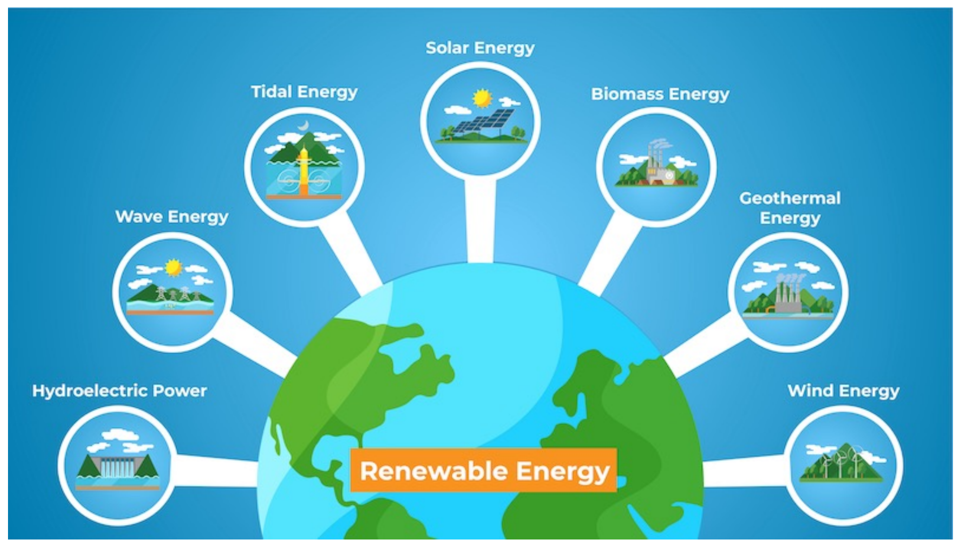




Credit: AMD CEO Lisa Su

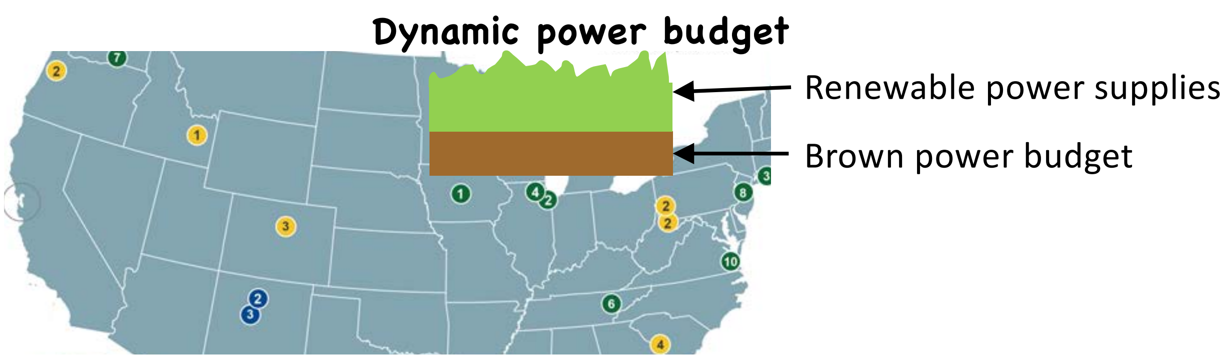
RMS in 2035

- Utilize **a diverse mix of power sources** (e.g., brown and green energy)
  - Operating leadership facilities solely on renewable energy is controversial
  - Collaboration with DOE EERE and other agencies is required



RMS in 2035

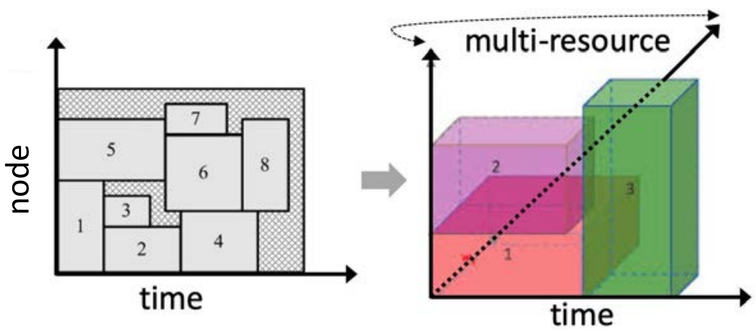
- **Power/energy as a schedulable resource**
- Scheduling, e.g., dynamic power budget**, across time and location**
    - o Time: Prioritizing jobs in varying orders based on power consumption
    - o Location: Allocating jobs to different resources for optimal energy efficiency



Challenges

➢ Managing heterogeneous resources among hybrid workloads is **a multi-dimensional combinatorial optimization problem**!

- From node-centric to multi-resource management
- Strategies for hard/soft constraints, e.g., on-demand and coupled
- Coordination across the components and the system stack
- Orchestration of computing and data movement
- Trade-offs (user vs facility, power vs performance)
- Rapid decision making in dynamic environment



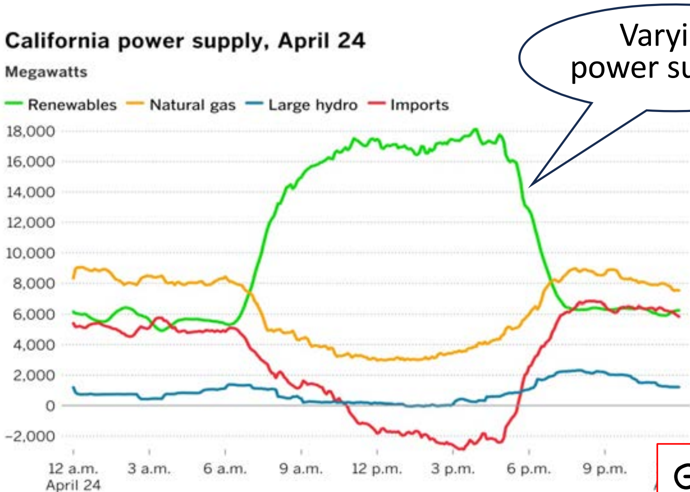
Challenges:

1. Multi-resource demand
2. Hard & soft constraints
3. Tunable shapes for some jobs
4. Adaptation to changes

Research Question

➢How can RMS manage **various dynamic factors** in concert?

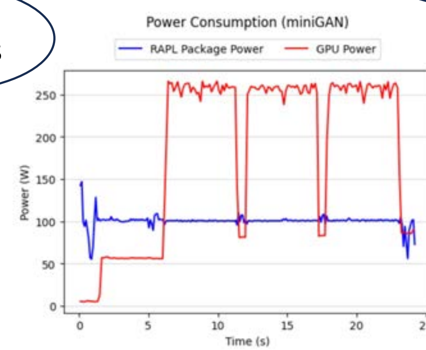
Varying
power supplies

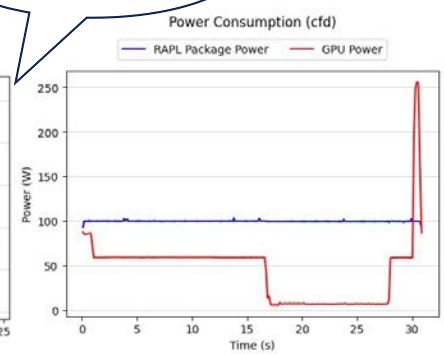


https://www.latimes.com/environment/newsletter/2021-04-29/solar-power-water-canals-california-climate-change-boiling-point

Dynamic
resources

Diverse & dynamic
power patterns



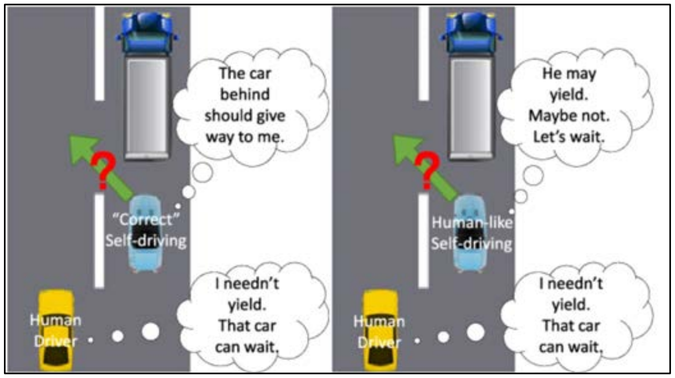


Gap: lack of a unified interface for power profiling across the system

Can RMS automatically learn the high-quality policies for energy efficiency specific to the computing environment?

Opportunity 1: Reinforcement Learning

Game playing

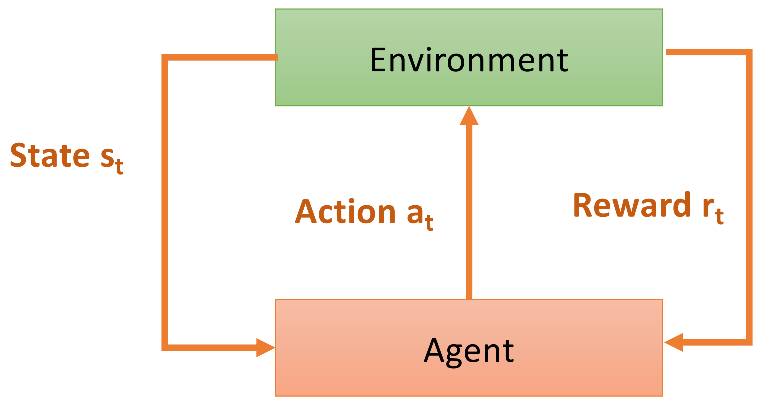


Self-driving cars

Robot control

Agent – learning optimal policies in a dynamic environment

Reinforcement Learning (cont.)



State s_t:

- Include both jobs & resources

Action a_t:

- Select a waiting job to run

Reward r_t:

- Reflect scheduling objective

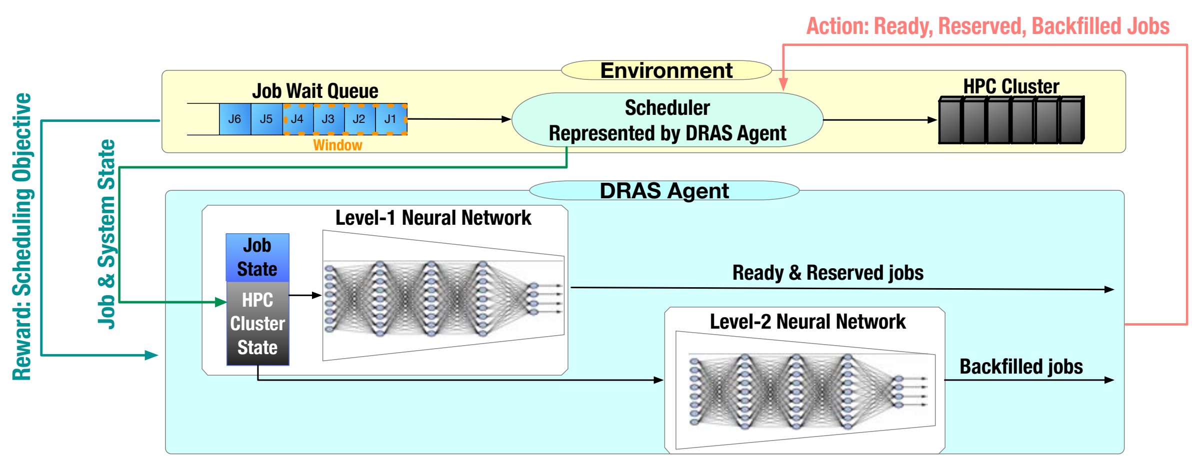
Goal: Maximize the cumulative reward $\sum_t r_t$

Features:

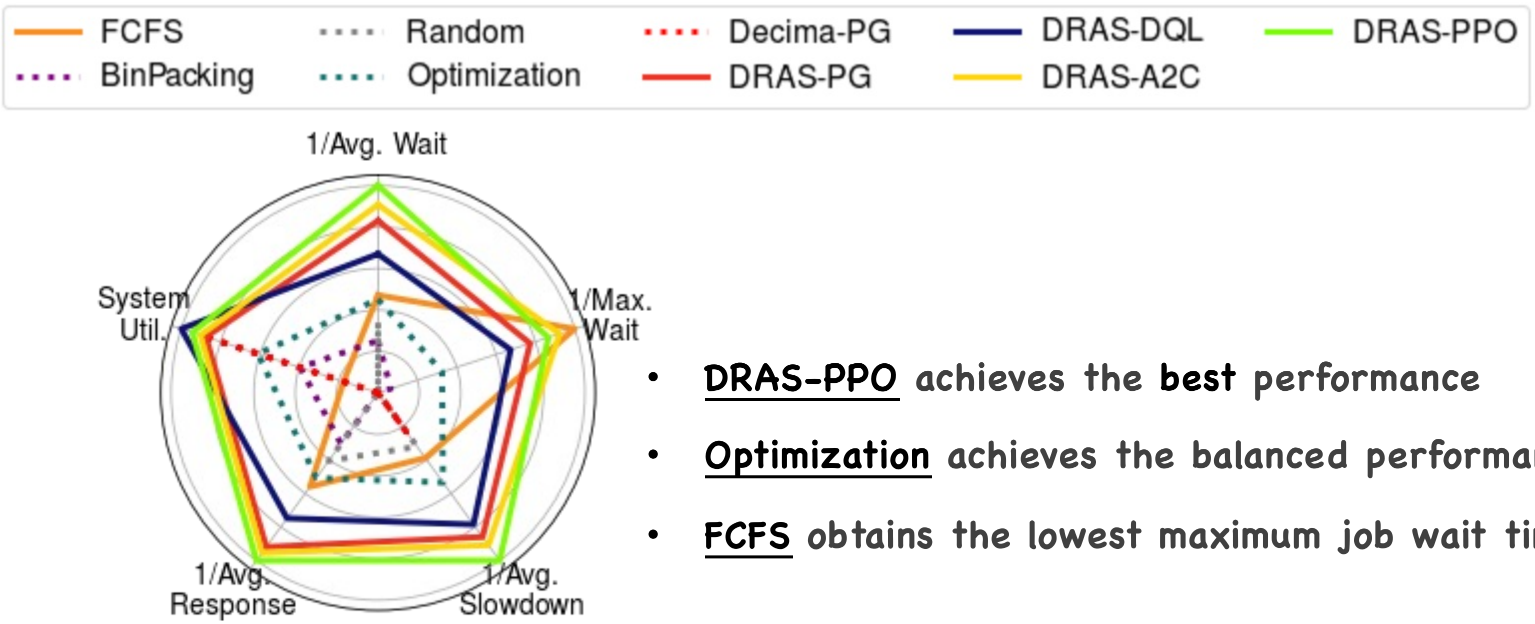
- Maximization of cumulative rewards over time, rather than just immediate gain
- Optimal decision making in dynamic environment
- Autonomous learning and adaptation
- …

Example

- ➢DRAS = Deep Reinforcement Agent for Scheduling
- ➢Key features
  - HPC domain knowledge, e.g., advanced reservation, backfilling
  - Multi-phase training strategy
  - Four RL algorithms: Deep Q-learning (DQL), Policy gradient (PG), Advanced actor critic (A2C), Proximal policy optimization (PPO)



Example (cont.)

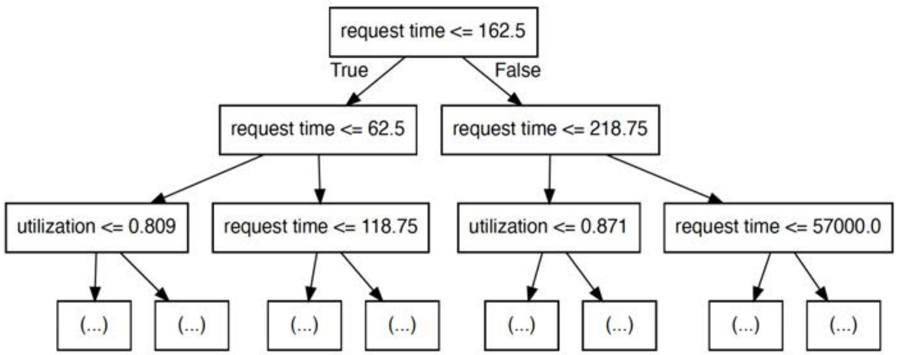


- **DRAS-PPO** achieves the **best** performance
- **Optimization** achieves the balanced performance
- **FCFS** obtains the lowest maximum job wait time

Y. Fan, et al., "DRAS: Deep Reinforcement Learning for Cluster Scheduling in High Performance Computing", ***IPDPS 2021*** *& IEEE Trans on Parallel and Distributed Systems* (***TPDS*** **2022**).

Example (cont.)

- Promising, but …
  - DNN is a blackbox
  - Hard to comprehend, debug, and adjust in practice
- Can we develop interpretable RL scheduling (IRL) model?



Interpretable tree generated from DNN model of 1,664-node DataStar

- ✓ IRL ≈ black-box DNN
- ✓ IRL faster decision:
  - IRL: 0.3ms per job selection
  - DNN: 20ms per job selection

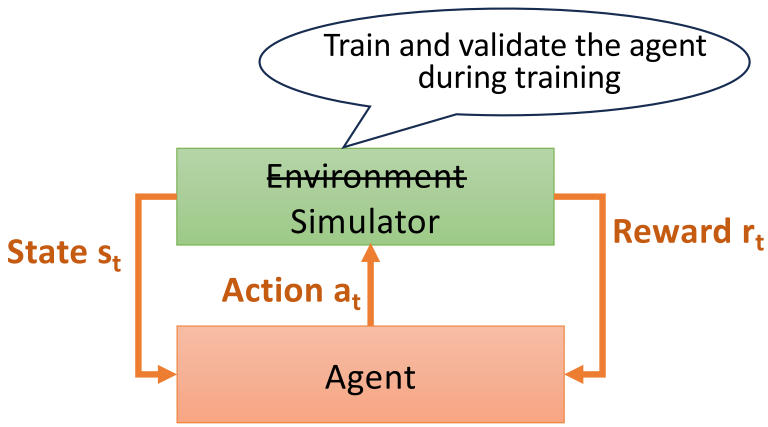
B. Li, et al., “Interpretable Modeling of Deep Reinforcement Learning Driven Scheduling”, ***MASCOTS 2023***.

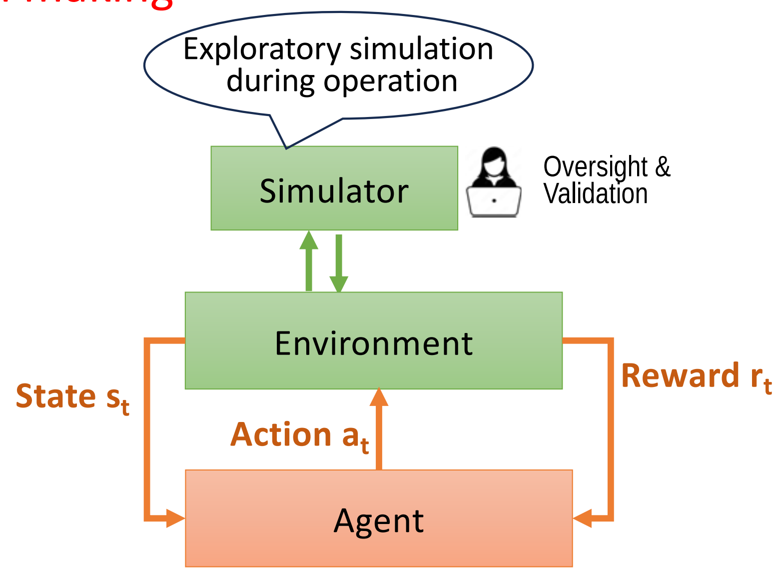
RL for RMS

- Reinforcement learning (RL) shows promising for RMS
  - + Autonomous decision-making in dynamic environments
  - + Plan-ahead focused on long-term rewards
- However, many challenges remain!
  - Problem formulation
  - Stability and convergence
  - State space explosion
  - interpretabilit
  - Training cost and inference latency
  - ...

Opportunity 2: Simulator

- Simulators are typically used for offline what-if analysis
- Time to harness simulator for dynamic decision making





- Explore exploratory simulation for (near) real-time what-if analysis
- Challenges: (1) power modeling, and (2) trade-off between accuracy and time overhead

Takeaway

- RMS plays a crucial role in EECS
  - Vertical coordination between applications and resources
  - Horizontal coordination among resources at all levels

- Lots of knobs & dynamic factors

- Intelligent resource management
  - Leverage reinforcement learning and other AI/ML technologies
  - Integrate exploratory simulation

- Many challenges and opportunities remain