ASIA SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE

DRL-Cloud: Deep Reinforcement Learning-Based Resource Provisioning and Task Scheduling for Cloud Service Providers

Mingxi Cheng Duke University, Durham, NC, USA

Ji Li, and Shahin Nazarian USC, Los Angeles, CA, USA

Speaker: Shahin Nazarian

USC University of Southern California

Duke UNIVERSITY

Outline

- Introduction
- System Model
  - User-workload, cloud platform, energy consumption, and realistic pricing
- DRL-Cloud
  - Task decorrelation
  - Two-stage resource provisioning (RP) and task scheduling (TS)
    - Control algorithm
    - Deep Q-learning RP-TS algorithm
- Results
- Conclusion

Introduction

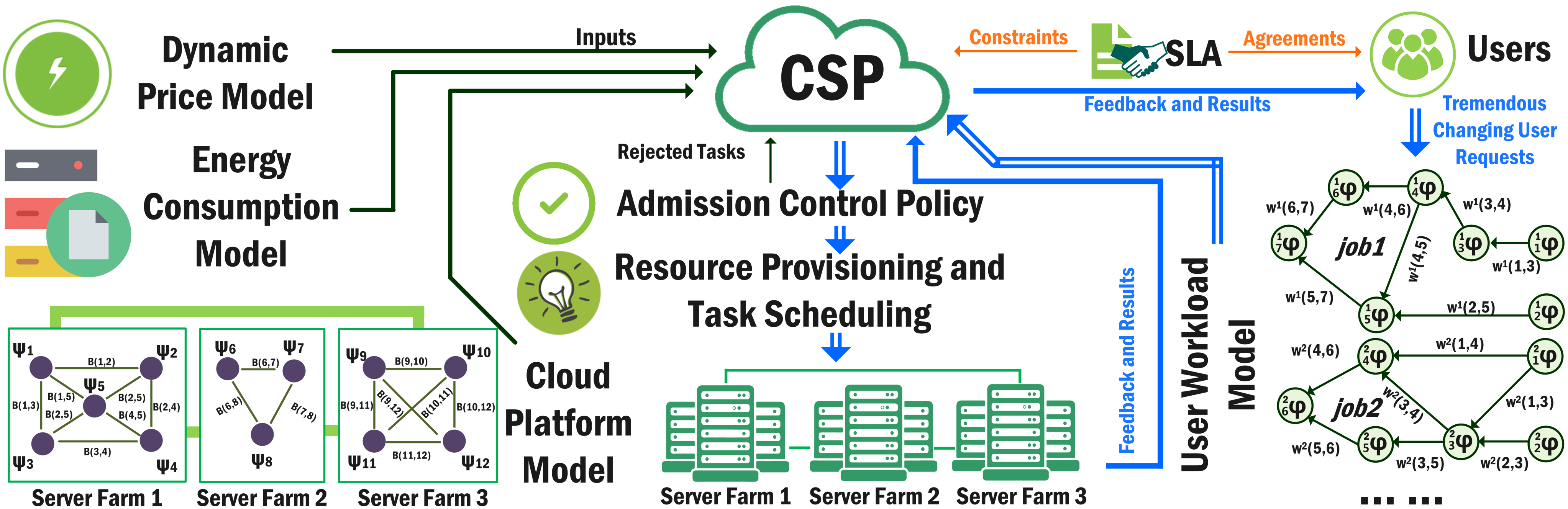
- **Cloud Computing**
  - **Omnipresent and on-demand access**
  - **Shared pool of configurable resources**
  - **Virtualization technology**
- **Roles**
  - **Cloud Service Providers (CSPs)**
  - **Users**
- **Examples of well-known CSPs**
  - **Google App Engine (GAE)**
  - **Amazon Elastic Compute Cloud (EC2)**

Introduction

- **Challenges**
  - **Tremendous energy costs in terms of electricity**
    - **E.g., data center electricity consumption projected to be ~140 billion KWh annually by 2020, i.e., ~13 billion US dollars annually in electric bills**
- **Goals**
  - **Reduce both energy consumption and energy cost in terms of electricity**
  - **Increase the profit margin of large-scale CSPs, and as well reduce the carbon footprint**

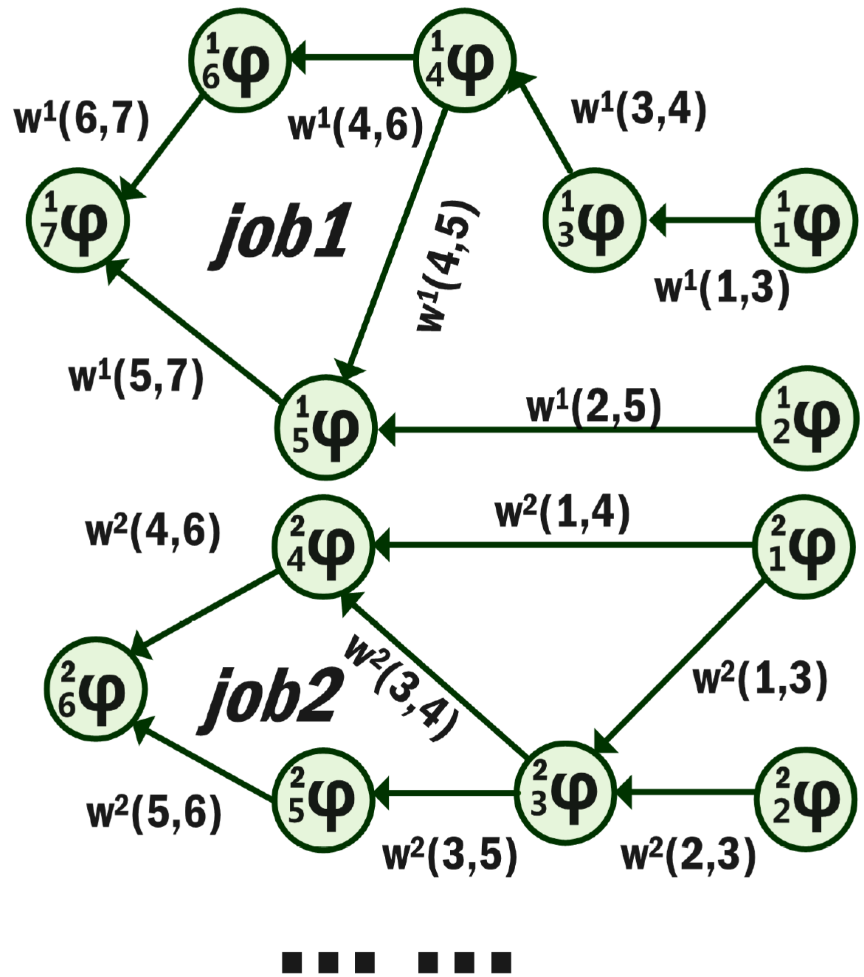
System Model

- **Consists of the models for:**
  - **user workload**
  - **cloud platform**
  - **energy consumption**
  - **realistic pricing**



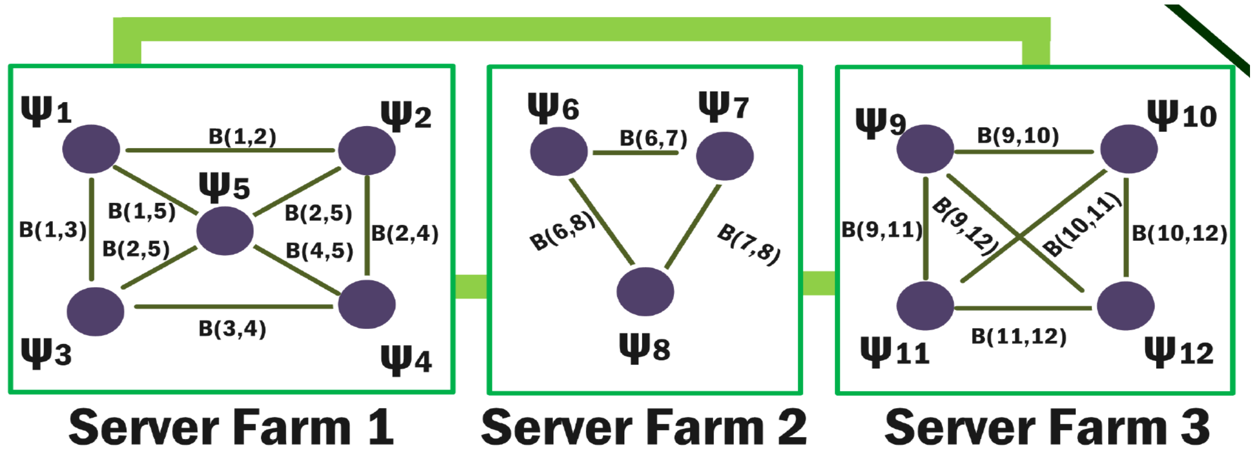
User Workload Model

- **Job characteristics:**
  - Directed Acyclic Graphs (DAGs) are used to model jobs
  - Vertex (φ): a single task
  - Edge (w): the amount of data that needs to be delivered



- **Task characteristics:**
  - Requested VM types
  - Estimated executable time
  - Deadline
  - Required amount of CPU and memory

Cloud Platform Model



- The cloud platform is modeled as an undirected graph:
  - Vertex (ψ): a server
  - Edge (B): the communication channel
- Server farm:
  - Nearby servers are clustered in a server farm
  - Servers farms are connected with each other through two-way high-speed channels
  - Servers within one server farm are connected through local channels
- Server:
  - Available CPU and Memory

Energy Consumption Model & Realistic Price Model

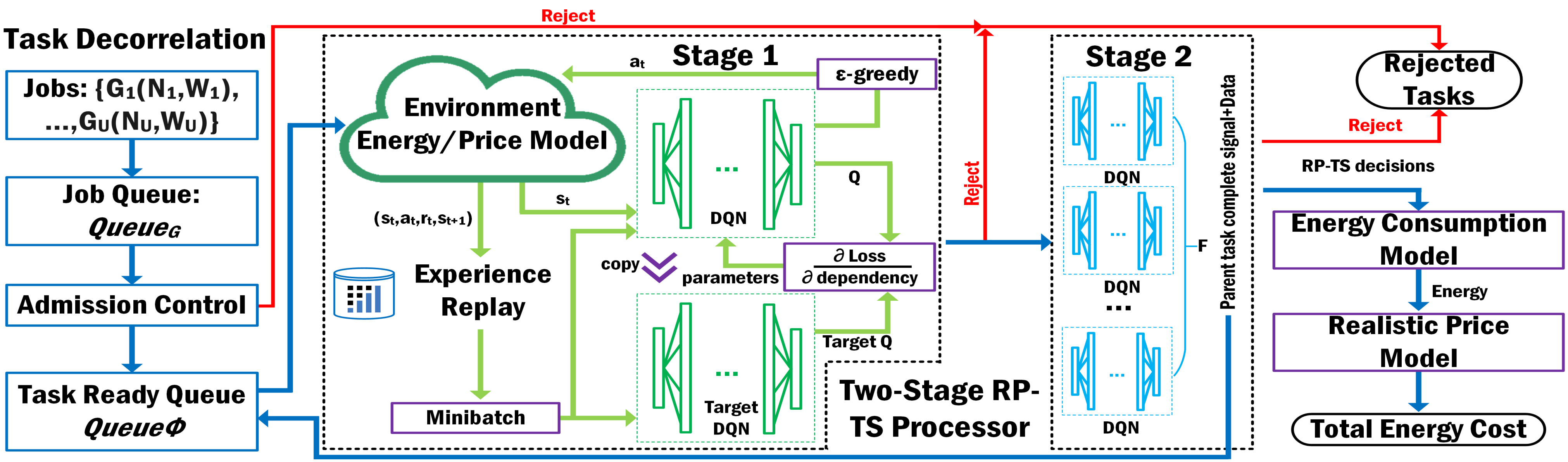
- **Utilization rate of server m at time t:** $\boldsymbol{Ur^{m}(t)}$
- **Total power at time t:**
  - $\boldsymbol{Pwr_{ttl}(t) = \sum_{m=1}^{M} Pwr_{st}^{m}(t) + Pwr_{dy}^{m}(t)}$
- **Total cost is based on TOUP(t) and RTP(power(t))**
  - $\boldsymbol{TotalCost = \sum_{t=1}^{T} Price(t, Pwr_{ttl}(t))}$

Problem Formulation

- ***Given*** **price, workload and cloud platform information**
- ***Find*** **the VM configuration and start time for each task**
- ***Minimize*** **the total energy price:**
  - $\boldsymbol{TotalCost = \sum_{t=1}^{T} Price(t, Pwr_{ttl}(t))}$
- ***Subject to*** **various constraints**

DRL-Cloud Framework: Resource Provisioning (RP) and Task Scheduling (TS) based on Deep Reinforcement Learning

- **Task decorrelation:**
  - **User request acceptance and decoupling**
  - **Construct job queue and task ready queue**
- **Two-stage RP-TS processor:**
  - **Energy cost minimization**



Architecture

- **Stage 1:**
  - **Allocate task to one server farm**
  - **Determine task start time**
  - **Continue processing or drop the job if SLA is violated**
- **Stage 2:**
  - **Choose the exact server to run task**
  - **Send parent task complete signal and data to job queue when task is completed**

Algorithm 1: Control Algorithm for DRL-Cloud

Initialize realistic price model
Initialize environment and deep Q-network DQN_{Stage_1}
Run DQN_{Stage_1} and store user request allocation
Initialize environment and deep Q-network DQN_{Stage_2}
for $f = 1, F$ **do**
    for $\tau = 1, T$ **do**
        Run DQN_{Stage_2} and store user request allocation
    end
end
Calculate final user request allocation matrix, i.e., Ur for every server
Calculate final energy consumption Pwr_{ttl} and electric bill $TotalCost$
return $TotalCost$

Algorithm 2: Deep Q-Learning for DRL-Based RP-TS With Experience Replay

Initialize replay memory Δ to capacity Ω
Initialize action-value function Q with random weights δ
Initialize target action-value function $\hat{Q}$ with weights $\delta' = \delta$
for $episode = 1, E$ **do**
    Reset cloud server environment to initial state
    Initialize sequence $s_1 = \{x_1\}$
    for $t = 1, T$ **do**
        With probability ϵ choose a random action a_t
        otherwise choose $a_t = argmax_a Q(s_t, a; \delta)$
        Execute action a_t and observe next observation x_{t+1}, reward r_t, and reject signal
        if $reject = 1$ **then**
            Run DQN again to get a new action a'_t
            if $a_t \neq a'_t$ **then**
                Replace a_t with a'_t
            end
        end
        Set $s_{t+1} = s_t, a_t, x_{t+1}$
        Store transition (s_{t+1}, a_t, r_t, s_t) in Δ
        Sample random minibatch of transitions (s_{j+1}, a_j, r_j, s_j) from Δ
        $target_j = \begin{cases} r_j, \text{if episode terminates at step j+1} \\ r_j + \gamma max_{a'} \hat{Q}(s_{j+1}, a'; \delta'), \text{otherwise} \end{cases}$
        Perform a gradient descent step on $\left(target_j - Q(s_j, a_j; \delta)\right)^2$
        Every Γ steps, train evaluation network, decrease ϵ
        Every ζ steps, copy Q to $\hat{Q}$
    end
end
return All actions a_t

Training Details

- **Action space:**
  - **Stage 1: {$\text{farm}_1\text{time}_1$,…,$\text{farm}_F\text{time}_T$}**
  - **Stage 2: {server_1,…,server_M}**

- **State space:**
  - **The optimal action determined based on current observation, which is the combination of current server observation x_{server} and current task observation x_{task}**
  - **x_{server}: available CPU and memory of requested VMs on server**
  - **x_{task}: requested CPU and memory, and task deadline**

Training Details

- **Reward function:**
  - **After taking action at at current state, system will evolve into a new state and receives a reward from the environment**
  - **Which is energy cost increase of action, i.e., current energy cost minus previous energy cost**

- **Training details:**
  - **Experience replay**
  - **Target network**
  - **Exploration & Exploitation**

Experiment Setup – Baselines

- **Greedy Baseline: CSP tries every option to find the assignment that yields the minimum energy cost increase. Also rejects tasks according to SLA**
- **Round-Robin (RR) Baseline: The CSP assigns each task in circular order. If the current assignment violates SLA, the scheduler will try the following options until nonviolation. A task will be rejected by the CSP if it is rejected by all possible assignments**
- **FERPTS* Baseline: one contemporary algorithm, that is aware of historical decisions and current scheduling of other tasks with the introduction of congestion concept**

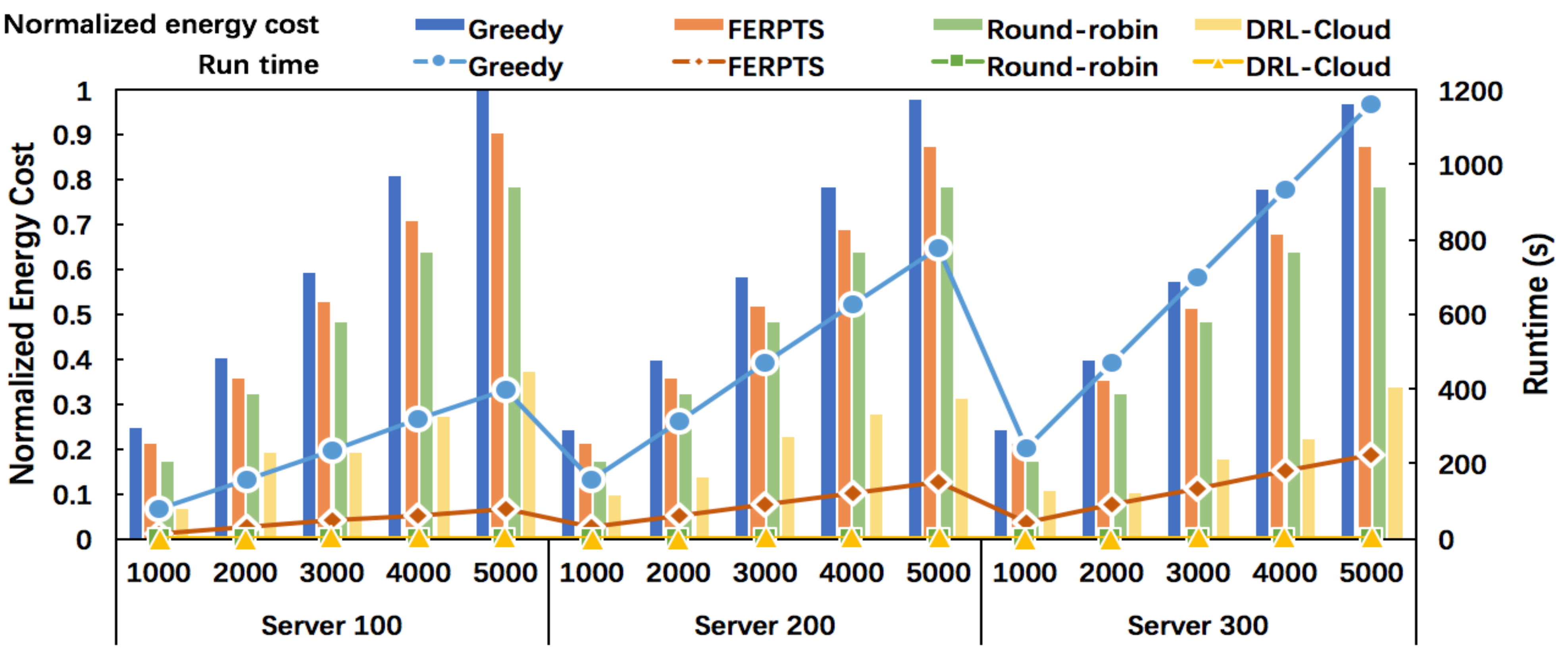
* H. Li et al., “Fast and energy-aware resource provisioning and task scheduling for cloud systems,” ISQED 2017

Experiment Setup

- **Small-scale:**
  - **3000 to 5000 user requests**
  - **100 to 300 servers that are clustered into 10 server farms**
- **Large-scale:**
  - **50000 to 200000 user requests**
  - **500 to 5000 servers that are clustered into 10 to 100 server farms**
- **User request are retrieved from Google cluster usage traces (29 days)**
- **Learning rate 0.1 , discount factor 0.9 in deep Q-learning**

Experiment Results – Small-Scale

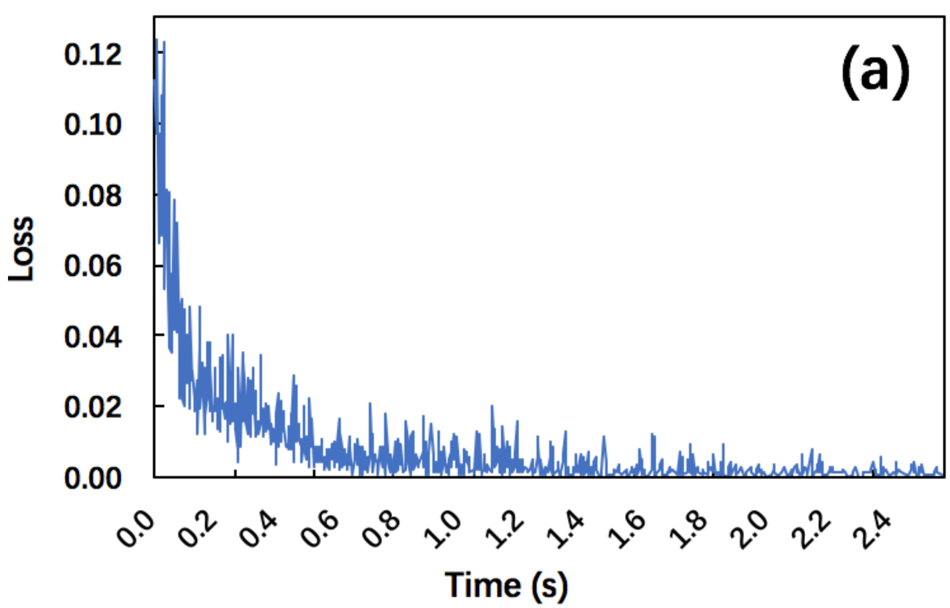
- **Up to 3X energy cost and 92X runtime reduction (compared to FERPTS)**

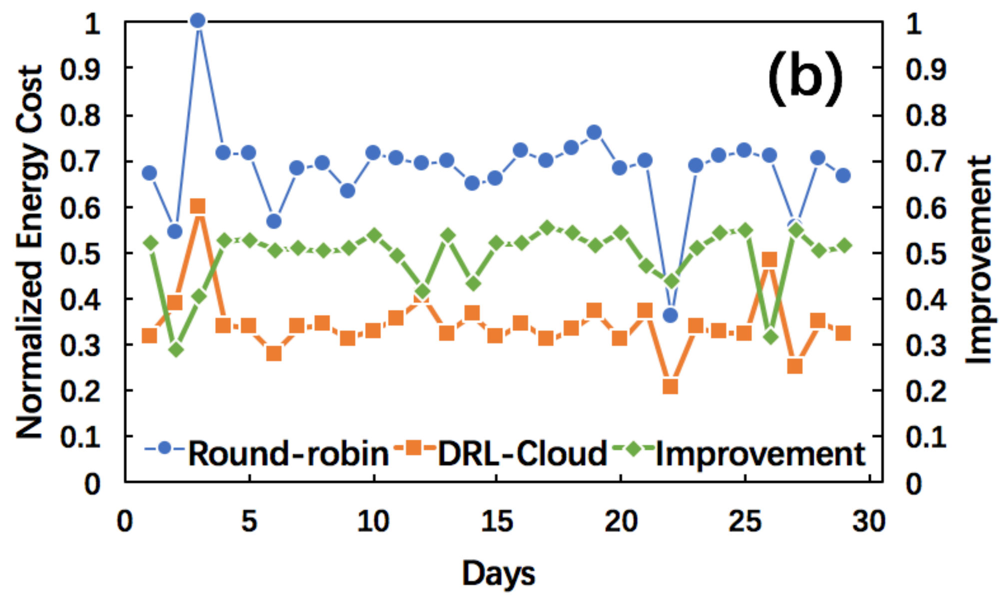


- **Runtime & energy cost comparisons with baselines for small-scale**
- **Energy cost is normalized w.r.t. that of Greedy for 100 servers & 5000 requests**

Experiment Results – Large-Scale, Long Term

- **2X energy cost efficiency improvement, 2X runtime reduction and 3X reject rate reduction on average (compared to RR)**





(a) Convergence of DRL-Cloud

(b) Energy cost comparison with RR in long-run (29 days) on large scale workloads and platform configuration (5000 servers and 50000 tasks)

Conclusion

- **DRL-Cloud a two-stage resource provisioning and task scheduling to reduce energy cost for cloud service providers with large-scale data centers and large amounts of user requests with dependencies.**
- **DRL-Cloud is highly scalable and highly adaptable compared to the state-of-the-art methods, and the training algorithm converges fast.**
- **Results of DRL-Cloud were compared to the baseline methods and demonstrated significant energy cost and runtime savings.**