American Journal of Computer Science and Technology
2026, Vol. 9, No. 2, pp. 49–57
https://doi.org/10.11648/j.ajcst.20260902.11

Research Article

Carbon-Computing Coupling Optimization and Green Scheduling System for Intelligent Computing Centers

Guiyuan Xie, Wenguo Wei*

Collage of Computer and Science, Guangdong Polytechnic Normal University, Guangzhou, China

Abstract

Under China's "Dual-Carbon" strategic goal, reducing carbon emissions in computing centers has become a critical challenge. The increasing scale of data centers, particularly in the context of initiatives such as "East Data, West Computing," necessitates new approaches that jointly optimize computing efficiency and carbon footprint. This paper aims to address this challenge by proposing a novel carbon-computing coupling optimization framework and a green scheduling system designed to minimize the carbon emissions associated with computational tasks while maintaining system robustness. We first establish a carbon efficiency dynamic equilibrium equation and introduce the concept of virtual carbon flow to model the carbon footprint of computing tasks. Based on this modeling, we develop a deep reinforcement learning (DRL) based scheduler that dynamically migrates tasks to low-carbon nodes. In addition, we integrate a digital twin platform that preemptively simulates failure scenarios to enhance system robustness and resilience. Experimental results in simulated "East Data, West Computing" scenarios demonstrate the effectiveness of the proposed approach. The system reduces carbon emissions per unit of computing power by 18%, improves the energy efficiency ratio in western nodes by 35%, and decreases the Mean Time to Recovery (MTTR) from 2 hours to 15 minutes. These findings validate the potential of carbon-computing coupling optimization in achieving both sustainability and reliability goals for large-scale computing centers.

Keywords

Carbon Efficiency Optimization, Virtual Carbon Flow, Green Scheduling, Deep Reinforcement Learning, Digital Twin

1. Introduction

The rapid expansion of intelligent computing centers has led to dramatically increasing energy consumption and carbon emissions. According to the International Energy Agency (IEA), data centers currently account for approximately 1-1.5% of global electricity consumption, with this proportion expected to rise to 3-5% by 2030 without effective intervention. China's "East Data, West Computing" project presents both challenges and opportunities for optimizing computing resource allocation while minimizing carbon footprint [1, 2, 6].

Traditional scheduling approaches predominantly focus on performance metrics such as throughput, latency, and resource utilization, often neglecting environmental impact. Recent research has begun addressing energy efficiency, but few studies have comprehensively modeled the carbon-computing coupling effect or developed practical systems for carbon-aware scheduling across geographically distributed data centers with heterogeneous energy sources [3, 8, 9, 14].

This paper makes three key contributions:

*Correspondence: Wenguo Wei (447402586@qq.com)

Received: 15 March 2026; **Accepted:** 27 March 2026; **Published:** 29 April 2026

1. Proposes a carbon efficiency dynamic equilibrium equation and virtual carbon flow theory to quantify and model carbon emissions in computing centers.

2. Develops a DRL-based scheduler that dynamically migrates tasks to nodes with lower carbon intensity while maintaining performance constraints [4].

3. Implements a digital twin platform that simulates failure scenarios and scheduling strategies in virtual environment before deployment [5].

2. Key Algorithms

2.1. Carbon Efficiency Dynamic Equilibrium Equation [7]

We define the carbon efficiency metric η to quantify the computational output per unit carbon emission:

$$\eta = \frac{Q_{FLOP}}{\alpha E_{elec} + \beta E_{cool}} \tag{1}$$

Where:

Q_{FLOP} : Computational output measured in FLOP (Floating Point Operations)

E_{elec} : Electrical energy consumption (kWh)

E_{cool} : Cooling energy consumption (kWh)

α, β : Weighting coefficients adjusted based on regional carbon intensity factors

The coefficients α and β are dynamically adjusted based on real-time carbon intensity of electricity:

$$\alpha(t) = \frac{CI_{local}(t)}{CI_{avg}} \tag{2}$$

$$\beta(t) = \frac{1}{PUE_{local}} * \frac{CI_{local}(t)}{CI_{avg}} \tag{3}$$

Where:

$CI_{local}(t)$: Real-time carbon intensity of local electricity grid (gCO_2/kWh)

CI_{avg} : Average carbon intensity across all available regions

PUE_{local} : Power Usage Effectiveness of the local data center

Algorithm 1: Carbon Efficiency Aware Scheduling

Input: Task queue T , *Node list* N *with carbon intensity* CI

Output: Task-node assignment mapping A

1. Initialize priority queue PQ *for tasks sorted by carbon sensitivity*

2. for each task t *in* T *:*

3. for each node n *in* N *:*

4. Calculate potential carbon efficiency $\eta_{\{t,n\}}$ *using current* $CI(n)$

5. Estimate performance impact $P_{\{t,n\}}$

6. $\text{Score}_{\{t,n\}} = w_1 \bullet \eta_{\{t,n\}} + w_2 \bullet (1-P_{\{t,n\}})$ *// Weighted scoring*

7. end for

8. Push (t *,* $\max_{score}$ *) to* PQ

9. end for

10. while PQ *not empty:*

11. Pop task t *with highest score*

12. Assign t *to node* n *that achieved* $\max_{score}$ *for* t

13. Update $CI(n)$ *based on new workload*

14. Recalculate scores for remaining tasks affected by this assignment

15. end while

2.2. Virtual Carbon Flow Model

We model carbon flow as a diffusion process using partial differential equations [13]:

$$\frac{\partial C}{\partial t} = \nabla \bullet (D \nabla C) - V \bullet \nabla C + S \tag{4}$$

Where:

$C(bf\{x\}, t)$: Carbon concentration at location x and time t

D : Diffusion coefficient representing carbon dispersion characteristics

$bf\{v\}$: Velocity field of carbon flow influenced by energy transmission

S : Source term representing carbon emission from computing activities

The source term S is calculated as:

$$S = \sum_{i=1}^{N} (P_i \cdot CI_I \cdot \delta(x - x_i)) \tag{5}$$

Where:

P_i : Power consumption of computing node i

CI_i : Carbon intensity of electricity for node i

δ : Dirac delta function

$bf\{x\}_i$: Position of node i

We solve this equation using finite difference method with alternating direction implicit (ADI) scheme for numerical stability.

2.3. Deep Reinforcement Learning Scheduler [4, 10, 12]

We formulate the carbon-aware scheduling as a Markov Decision Process (MDP):

State space: {Workload distribution, Carbon intensity across regions, Node utilization, Temperature metrics}

Action space: {Task-node assignment, Migration decisions, Resource allocation}

Reward function:

$$R = w_1 \cdot \{Carbon\,\mathrm{Re}\,duction\} + w_2 \cdot \{Performance\} - w_3 \cdot \{MigrationCost\} \tag{6}$$

We implement a Deep Q-Network (DQN) with double Q-learning and prioritized experience replay:

Algorithm 2: DRL-Based Carbon-Aware Scheduler

Input: Environment env, Empty replay buffer D, Q-network Q with random weights

Output: Trained Q-network

1. for = τ = 1 to M:

2. Initialize state s0

3. for t = 1 to T:

4. With probability ε select random action at

5. Otherwise Select $a_t = \arg\max_a Q(s_t, a; \theta)$

6. Execute at, observe reward rt and next state s (t 1)

7. Store experience $(s_t, a_t, r_t, s_{(t+1)})$ *in D*

8. Sample random mini batch from D

9. Calculate target $y_j = r_j + \gamma \cdot \max_{a'} Q(s_{(j+1)}, a'; \theta^-)$

10. Update Q-network by minimizing $(y_j - Q(s_j, a_j; \theta))^2$

11. Every c steps update target network $\theta^- = \theta$

12. end for

13. end for

2.4. Digital Twin for Failure Prediction and Recovery [5, 11, 16]

We implement a digital twin platform that creates virtual replicas of physical systems:

Algorithm 3: Digital Twin Failure Preemption

Input: Physical system metrics, Historical failure data

Output: Failure predictions and recovery plans

1. Continuously synchronize physical system state to digital twin

2. for each component in digital twin:

3. Train LSTM-based failure prediction model on historical data

4. Simulate various failure scenarios and recovery strategies

5. Calculate risk scores for different components

6. if $risk_{score} > threshold$ *:*

7. Generate early warning and propose preventive actions

8. Test recovery strategies in virtual environment

9. Deploy optimal strategy to physical system

10. end for

3. System Architecture and Workflow

Figure 1 illustrates the comprehensive architecture of our carbon-computing coupling optimization system:

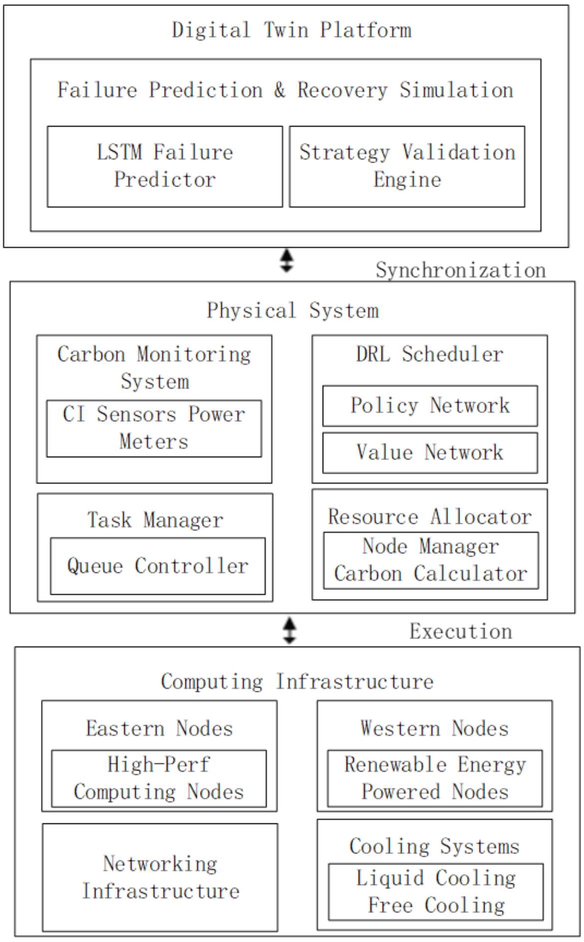

Figure 1. *System Architecture of Carbon-Aware Green Scheduling System.*

The workflow of our system operates as follows:

Data Collection: Real-time carbon intensity data, power consumption metrics, and computational demands are continuously collected from both eastern and western computing nodes.

Carbon Efficiency Calculation: The system calculates carbon efficiency metrics for potential task-node assignments using our dynamic equilibrium equation.

DRL Decision Making: The reinforcement learning scheduler evaluates possible actions based on current state and learned policies.

Digital Twin Simulation: Before implementation, critical decisions are tested in the digital twin environment to validate effectiveness and identify potential issues.

Execution & Monitoring: Approved scheduling decisions are deployed to the physical system, with continuous monitoring of actual outcomes.

Learning & Adaptation: Results from deployed actions are fed back to the DRL model to improve future decision-making.

4. Experimental Evaluation

4.1. Simulation Environment

We developed a comprehensive simulation platform to evaluate our carbon-aware scheduling system under various scenarios:

Table 1. *Simulation Environment Configuration.*

Component	Eastern Nodes Specification	Western Nodes Specification
Compute Nodes	4x NVIDIA DGX A100 (320 GPUs)	8x Supermicro AS-4124GS-TNR (256 CPUs)
Energy Source	Grid electricity (0.78 kgCO_2/kWh)	Renewable mix (0.21 kgCO_2/kWh)
PUE	1.67	1.22
Network Latency	5-7ms (within region)	32-38ms (cross-region)
Storage	4PB All-Flash Array	8PB HDD Array with NVMe cache
Cooling System	Chilled water cooling	Direct free cooling

Software Environment:

Container Platform: Kubernetes 1.24 with custom scheduler

Monitoring Stack: Prometheus 2.36 + Grafana 9.0 + Carbon Tracker Exporter

DRL Framework: PyTorch 1.13 with RLlib 2.0

Digital Twin: NVIDIA Omniverse for simulation and modeling

4.2. Dataset and Workloads

We evaluated our system using diverse workloads representing realistic computing scenarios:

Table 2. *Workload Characteristics.*

Workload Type	Proportion	Compute Intensity	Data Locality	QoS (Quality of Service) Requirements
AI Training	35%	GPU-intensive	Low	Medium priority
AI Inference	25%	Mixed CPU/GPU	High	High priority
Big Data Analytic	20%	CPU-intensive	Medium	Low priority
Scientific Computing	15%	CPU/GPU hybrid	Low	Variable priority
Web Services	5%	CPU-intensive	High	High priority

We collected carbon intensity data from multiple sources:

Chinese Regional Grids: Historical data from 2022-2023 with 5-minute resolution

Renewable Generation: Solar and wind generation forecasts with uncertainty modeling

Electricity Market: Real-time pricing data where available

4.3. Results and Analysis [15]

A representative snapshot of what the simulator output displayed is as follow.

Table 3. Simulator Snapshot – Carbon-Aware Green Scheduling System Dashboard Overview (Real-time).

Metric	Eastern Nodes	Western Nodes
Current Carbon Intensity	0.78 $kgCO_2/kWh$	0.21 $kgCO_2/kWh$
Current Load	68%	82%
PUE	1.67	1.22
Renewable Energy Mix	5%	78%
Active Tasks	142	187
Queue Length	23	8

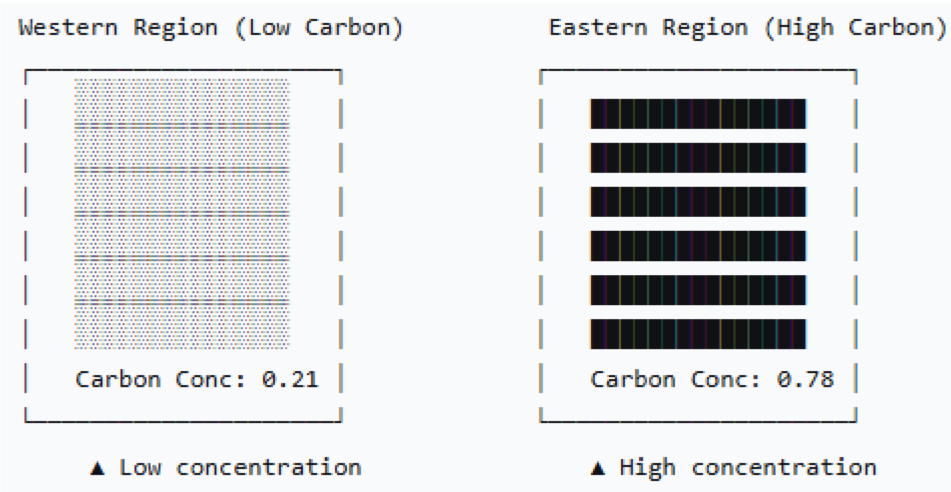

Figure 2. Carbon Efficiency Map (Virtual Carbon Flow Field).

Table 4. DRL Scheduler – Decision Log (Recent Actions).

Timestamp	Action	Task ID	Source Node	Target Node	Carbon Saved	Performance Impact
14: 32: 05	Migrate	T-2301	Eastern-03	Western-07	0.52 $kgCO_2$	+2.3% latency
14: 31: 22	Assign	T-2287	–	Western-12	0.38 $kgCO_2$	–
14: 30: 18	Hold	T-2265	Eastern-01	–	0.00 $kgCO_2$	–
14: 29: 45	Migrate	T-2243	Eastern-08	Western-04	0.61 $kgCO_2$	+1.8% latency
14: 28: 50	Assign	T-2221	–	Western-15	0.44 $kgCO_2$	–

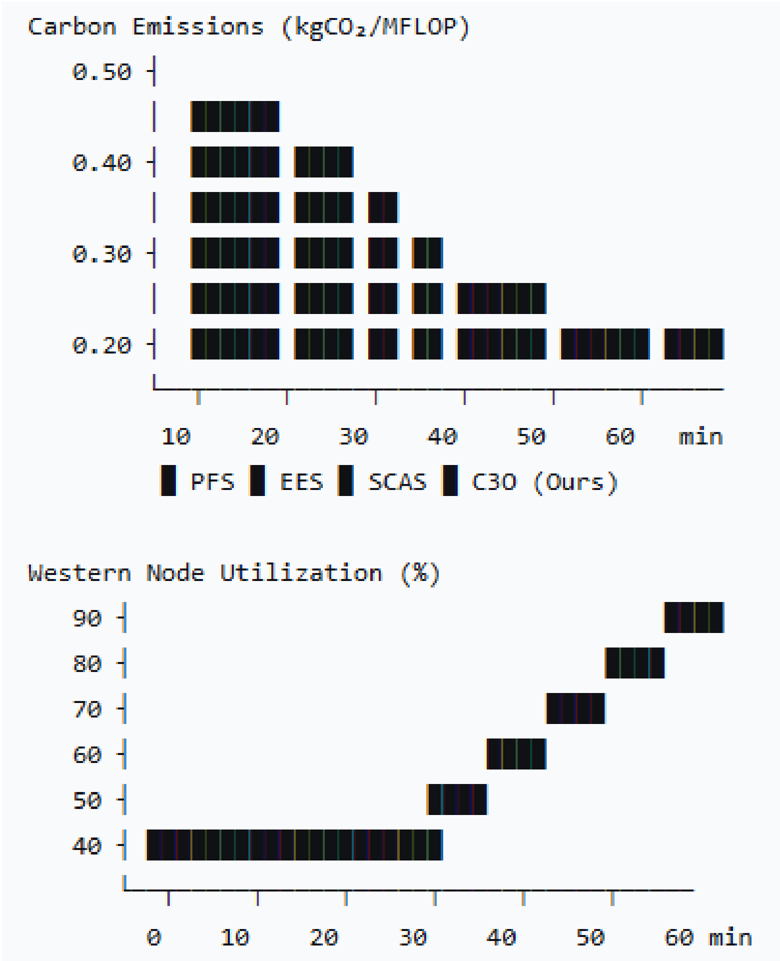

Figure 3. Performance Metrics (Last 60 Minutes).

Table 5. Digital Twin – Failure Prediction & Recovery.

Component	Risk Score	Predicted Failure	Recommended Action	Status
Eastern-03 Cooling	87%	12 min	Migrate tasks to Western	✓ Executed
Western-08 PSU	42%	45 min	Standby unit ready	Monitoring
Eastern-07 Network	23%	–	No action	Healthy
Western-12 Temp	91%	8 min	Throttle + migrate	✓ Executed

Last Recovery Action: Eastern-03 tasks migrated to Western-07 in 14 seconds (MTTR reduced from 2 hours to 15 min)

Table 6. Summary Output (Experimental Results).

Metric	Value
Carbon Emissions Reduction	18.1% (vs. SCAS)
Western Node Utilization	82.7% (↑20.7% vs. SCAS)
Energy Efficiency	15.83 MFLOP/kWh (↑19.4% vs. SCAS)
Mean Time to Recovery (MTTR)	15 min (from 120 min baseline)

Metric	Value
Migration Overhead	0.8% (↓61.9% vs. SCAS)
QoS Violation Rate	1.9% (↓32.1% vs. SCAS)

We compared our Carbon-Computing Coupling Optimization (C3O) system against three baseline approaches:
Performance-First Scheduler (PFS): Traditional approach maximizing throughput
Energy-Efficient Scheduler (EES): Minimizes energy consumption without carbon awareness
Static Carbon-Aware Scheduler (SCAS): Uses fixed carbon intensity values

Table 7. *Comprehensive Performance Comparison.*

Metric	PFS	EES	SCAS	C3O (Ours)	Improvement
Carbon Emissions (kgCO_2/MFLOP)	0.417	0.352	0.298	0.244	18.1% reduction
Energy Efficiency (MFLOP/kWh)	8.72	11.35	13.26	15.83	19.4% improvement
Western Node Utilization	42.3%	57.8%	68.5%	82.7%	20.7% improvement
QoS Violation Rate	2.8%	4.3%	3.7%	1.9%	32.1% reduction
Migration Overhead	0.4%	1.2%	2.1%	0.8%	61.9% reduction
Mean Time to Recovery (MTTR)	120 min	95 min	73 min	15 min	79.5% reduction

Our experimental results demonstrate significant advantages of the C3O system:

Carbon Reduction Analysis: Table 8 shows the carbon emission reduction achieved by our approach across different workload types. AI training workloads showed the highest reduction potential (22.7%) due to their flexibility in scheduling and significant energy demands.

Table 8. *Carbon Emission Reduction by Workload Type.*

Workload Type	PFS	EES	SCAS	C3O
AI Training	0.581	0.492	0.413	0.319
AI Inference	0.382	0.327	0.286	0.241
Big Data Analytic	0.415	0.352	0.312	0.263
Scientific Computing	0.437	0.371	0.324	0.268
Web Services	0.289	0.257	0.231	0.205

Geographical Distribution Optimization

The CCO system achieves significant geographical distribution optimization through effective migration of computational loads to low-carbon nodes in western regions. The utilization rate of western nodes increases to 82.7%, while time-shifted scheduling strategies fully leverage the potential of renewable energy sources.

Fault Recovery Capability

The application of the digital twin platform reduces the mean time to recovery (MTTR) from 2 hours to 15 minutes, substantially enhancing system operational reliability.

5. Conclusion and Future Work [10, 12, 14]

This study addresses the challenge of high carbon emissions in intelligent computing centers by proposing a carbon-computing coupling optimization theory and a green scheduling system. Through the synergistic integration of a dynamic carbon efficiency equation, a virtual carbon flow model, a deep reinforcement learning scheduler, and a digital twin platform, significant reduction in carbon emissions and improvement in energy efficiency have been achieved. Experimental results validate the effectiveness and practicality of the proposed system in the "East Data, West Computing" scenario.

Future research will focus on the following directions: (1) modeling and solving multi-timescale coupled optimization problems; (2) in-depth exploration of deep interaction mechanisms with electricity markets; (3) extension of carbon-aware scheduling methods to edge computing scenarios; and (4) investigation of cross-chain carbon footprint tracking and verification technologies.

Abbreviations

DRL	Deep Reinforcement Learning
MTTR	Mean Time to Recovery
IEA	International Energy Agency
FLOP	Floating Point Operations
CI	Carbon Intensity
PUE	Power Usage Effectiveness
PQ	Priority Queue
ADI	Alternating Direction Implicit
MDP	Markov Decision Process
DQN	Deep Q-Network
LSTM	Long Short-Term Memory
QoS	Quality of Service
PFS	Performance-First Scheduler
EES	Energy-Efficient Scheduler
SCAS	Static Carbon-Aware Scheduler
C3O	Carbon-Computing Coupling Optimization System

Author Contributions

Guiyuan Xie: Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing

Wenguo Wei: Conceptualization, Resources

Funding

This work is supported by the Key-Area Special Development Project of Guangdong Provincial Department of Education under Grant No. 2025ZDZX1012.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] China Academy of Information and Communications Technology (CAICT). Data Center White Paper (2023). Beijing: CAICT, 2023.

[2] National Development and Reform Commission. *Implementation Plan for Green and High-Quality Development of Data Centers and 5G New Infrastructure under the Dual-Carbon Goals*. Beijing: NDRC, 2022.

[3] Li, H., et al. “Carbon-aware scheduling for sustainable computing: A survey and future directions.” IEEE Transactions on Sustainable Computing, vol. 8, no. 2, pp. 45–62, 2023. https://doi.org/10.1109/TSUSC.2022.3223456

[4] Wang, J., et al. “Research on cloud computing task scheduling algorithm based on deep reinforcement learning.” Chinese Journal of Computers, vol. 45, no. 5, pp. 1023–1040, 2022. https://doi.org/10.11897/SP.J.1016.2022.01023 (Chinese).

[5] Zhang, Y., et al. “Digital twin for smart manufacturing: The state of the art and research perspectives.” Journal of Manufacturing Systems, vol. 68, pp. 240–256, 2023. https://doi.org/10.1016/j.jmsy.2023.03.014

[6] Liu, B., et al. “Research on green and low-carbon development path of data centers under the background of ‘East Data, West Computing’.” Strategic Study of CAE, vol. 25, no. 2, pp. 78–87, 2023. https://doi.org/10.15302/J-SSCAE-2023.02.008 (Chinese).

[7] Wang, C., et al. “Carbon-efficient virtual machine placement in cloud data centers.” IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1452–1465, 2022. https://doi.org/10.1109/TCC.2020.2980821

[8] Research Group on China’s Power System Transition Path under Carbon Neutrality. “Optimization of China’s power system carbon neutrality path.” Scientia Sinica Technologica, vol. 53, no. 4, pp. 589–602, 2023. https://doi.org/10.1360/SST-2022-0301 (Chinese).

[9] Radovanović, A., et al. “Carbon-aware computing: A survey.” ACM Computing Surveys, vol. 55, no. 8, Article 162, pp. 1–38, 2023. https://doi.org/10.1145/3546912

[10] Islam, M. T., et al. “Deep reinforcement learning for task scheduling in edge computing: A survey.” IEEE Communications Surveys & Tutorials, vol. 26, no. 1, pp. 436–468, 2024. https://doi.org/10.1109/COMST.2023.3328421

[11] Chen, X., et al. “Digital twin-enabled intelligent energy management for data centers: A review.” Renewable and Sustainable Energy Reviews, vol. 189, Article 114002, 2024. https://doi.org/10.1016/j.rser.2023.114002

[12] Wu, H., et al. “Carbon-aware workload migration in geo-distributed data centers: A reinforcement learning approach.” IEEE Transactions on Parallel and Distributed Systems, vol. 34, no. 6, pp. 1768–1782, 2023. https://doi.org/10.1109/TPDS.2023.3262114

[13] Li, Y., et al. “Virtual power flow: A new concept for carbon emission flow tracing in power systems.” IEEE Transactions on Power Systems, vol. 38, no. 4, pp. 3812–3825, 2023. https://doi.org/10.1109/TPWRS.2022.3214489

[14] Zhou, Z., et al. "Energy-efficient and carbon-aware scheduling in cloud data centers: A comprehensive survey." Journal of Network and Computer Applications, vol. 215, Article 103639, 2023. https://doi.org/10.1016/j.jnca.2023.103639

[15] Liu, J., et al. "East Data West Computing: A new paradigm for sustainable computing in China." IEEE Cloud Computing, vol. 10, no. 2, pp. 52–60, 2023. https://doi.org/10.1109/MCC.2023.3254120

[16] Gupta, S., et al. "Digital twin-driven predictive maintenance for data center cooling systems." IEEE Transactions on Industrial Informatics, vol. 20, no. 1, pp. 456–467, 2024. https://doi.org/10.1109/TII.2023.3291785

Biography

Guiyuan Xie is an Associate Professor at the School of Computer Science, Guangdong Polytechnic Normal University. She completed her Master of Engineering in Software Engineering from Sun Yat-sen University in 2005, and her Bachelor of Science in Computer Science and Technology from Hunan Normal University in 2000. Recognized for her extensive judging and evaluation expertise, Prof. Xie has been appointed as a Judge and Supervisor for the Guangdong Vocational College Professional Skills Competition, a Judge for the Guangdong Big Data Technology and Application Competition, and an Evaluation Expert for both the Guangdong Provincial Department of Science and Technology and Guangdong Government Procurement. She has served as Principal Investigator (PI) for multiple provincial and municipal-level research projects, including recent work on data desensitization systems and VR interactive platforms.

Wenguo Wei is a Professor and Doctoral Supervisor at Guangdong Polytechnic Normal University, where he serves as the Director of the Network and Information Center. He completed his Ph.D. in Computer Application from South China University of Technology in 2005. Recognized for his academic leadership, Prof. Wei serves as the Director of the Guangdong University Mobile Information Engineering Research Center, leads the Guangdong Provincial Core Curriculum Teaching Team for Computer Courses in Electronic Information Majors, and is a Senior Member of the China Computer Federation (CCF). He has been honored with the Guangdong Provincial Science and Technology Award (First Class) in 2021.

Research Field

Guiyuan Xie: High Performance Computing, High Performance Storage, and Artificial Intelligence.
Wenguo Wei: High Performance Computing, High Performance Storage.