A Survey of Energy Efficiency for Kubernetes Clusters

1st Laharraf Mohamed
SEEDS, INPT
Rabat, Morocco
laharraf.mohamed@doctorant.inpt.ac.ma

2th Kamal Idrissi Hamza
SEEDS, INPT
Rabat, Morocco
kamalidrissi@inpt.ac.ma

3th Allaki Driss
SEEDS, INPT
Rabat, Morocco
d.allaki@inpt.ac.ma

2025 International Conference on Intelligent Systems: Theories and Applications (SITA) | 979-8-3315-5989-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/SITA67914.2025.11273401

***Abstract*—Kubernetes excels in providing performance and scalability for cloud-native applications, but its energy consumption is a significant concern regarding sustainability. The aim of this research is to address this issue by integrating energy efficiency into Kubernetes. We examine three fundamental areas. First, we discuss new methods in monitoring and measuring energy usage. Second, we examine current best practice systems for energy-conscious scheduling. Third, we discuss architectural patterns employed for efficient energy management. The survey discusses fundamental tools and techniques, identifies research and operation-facing challenges, and charts a path forward to attain sustainable, energy-conscious practices in cloud-native Kubernetes clusters.**

***Index Terms*—Cloud-Native Applications, Kubernetes, Energy Efficiency, Energy Monitoring, Resource Management, Energy-Aware Scheduling, Carbon-Aware Scheduling.**

I. Introduction

Despite its leadership in container orchestration, Kubernetes faces a growing challenge in energy efficiency. Achieving this requires balancing energy savings from over-provisioning against performance loss from under-provisioning [1]. This necessitates dynamic optimization via autoscaling [2], right-sizing [3], consolidation [4], and innovation in monitoring and scheduling [5].

In spite of this emphasis, current studies offer minimal in-depth information on energy optimization in Kubernetes. Previous research focus on different domains, like VM consolidation [6], overall cloud or container power consumption without significant Kubernetes details [7], non-orchestrated container performance [8], or certain algorithmic aspects in Kubernetes. Even pertinent research from domains like HPC [9], [10] is not applicable because of Kubernetes' distinct orchestration layer and microservice models. Thus, an actual gap exists here: there is no dedicated survey that appropriately addresses the intersection of optimization methods, metrics required, and multiple scheduling approaches adequate for Kubernetes energy efficiency.

This article specifically aims to cover this gap by providing a comprehensive rundown of where energy efficiency stands in Kubernetes deployments. This paper integrates current advancements, starting with a review of basic monitoring and measurement methods, tools, and challenges, which is then followed by a discussion of state-of-the-art energy-aware scheduling systems and architectural proposals. In summary, this work seeks to give both researchers and practitioners a detailed technical review, outline the current state of the art, highlight existing challenges, and identify potential directions for enhancing energy efficiency in Kubernetes.

II. Monitoring and Measurement of Energy Consumption

Optimizing Kubernetes energy efficiency relies on suitable monitoring and measurement [4]. Accurate energy data is key, providing cluster usage insights [11] to find inefficiencies and evaluate optimizations. This section discusses measurement needs, compares tools/models, explores granularity, and examines complexities in Kubernetes energy monitoring.

A. Energy Monitoring Frameworks and Tools

A variety of frameworks and tools are emerging to monitor energy consumption within Kubernetes, differing in methodology, data sources, detail level, and precision. One prominent example is *Kepler* (Kubernetes-based Efficient Power Level Exporter), a CNCF Sandbox project that estimates process-level power using eBPF and a ratio power model, exporting data as Prometheus metrics. However, its accuracy has been questioned in independent studies, particularly concerning how it assigns power to inactive containers and system processes [11], [21].

Another tool, *POET* (Platform for O-RAN Energy Efficiency Testing), focuses specifically on measuring and modeling Energy Efficiency (EE) and Energy Savings (ES) in Open Radio Access Network (O-RAN) 5G deployments running on Kubernetes. It integrates measurements from physical hardware and containerized network functions (CNFs/VNFs), using external devices for server-level power capture, highlighting the need for domain-specific approaches in areas like telecommunications [3], [12]. *Scaphandre* is an open-source tool utilizing RAPL (Running Average Power Limit) to measure application energy consumption, correlating CPU usage with energy draw to estimate power per process. When deployed as a DaemonSet in Kubernetes, it labels metrics with pod and namespace information. Its accuracy hinges on RAPL availability and the assumption linking CPU usage directly to power consumption [13], [14]. *KubeWatt* presents a different model, using the Redfish API for total node power measurement and Kubernetes metrics APIs for CPU utilization

data. It separates node power into static and dynamic parts, attributing the dynamic portion to containers based on their CPU usage. Research indicates that *KubeWatt* offers better accuracy than *Kepler*, notably by avoiding power attribution to idle containers [11]. Finally, *CEEMS* provides a monitoring stack designed for diverse environments, leveraging standard observability tools to gather data and report real-time energy use and estimated CO2 emissions. Its flexible architecture supports custom rules for estimating workload energy consumption [16].

The following table provides a comparative analysis of these key monitoring tools, summarizing their core attributes to guide selection for different use cases.

TABLE I
COMPARISON OF ENERGY MONITORING TOOLS

Tool	**Primary Technology**	**Measurement Granularity**	**Key Features**	**Primary Limitation(s)**
Kepler	Hardware Counters model, RAPL	Process, Container, Pod, Node	Prometheus Export, ML Model option, eBPF support	Model Accuracy, Hardware Dependency (for RAPL), Overhead.
POET	Hardware Sensors, Container Metrics	System, Container	O-RAN specific, Bare-metal + K8s integration	Specific Use Case (Telecom/O-RAN).
Scaphandre	RAPL, other sensors	Process, Host (Node)	Prometheus Export, Lightweight	Hardware Dependency, Attribution less granular.
KubeWatt	Typically RAPL or other hardware sensors	Node, potentially Container	Prometheus Export	Hardware Dependency, Granularity may vary.
CEEMS	Integrates underlying tools (e.g., Kepler/Scaphandre), RAPL	Container, Pod, Node	Prometheus/ Grafana Integration, Carbon Metrics	Depends on underlying collector accuracy, Setup Complexity.

The ongoing development and comparative analysis of these tools, particularly the contrasting approaches of *Kepler* and *KubeWatt* regarding container-level attribution, highlight a critical maturation phase in Kubernetes energy monitoring. The focus is shifting decisively from basic node-level measurements towards obtaining accurate, actionable data at the workload (container/pod) level.

B. Granularity and Challenges in Energy Measurement

Energy consumption in Kubernetes can be measured or estimated at various levels of granularity, each presenting distinct advantages and challenges:

Process-level: Offers the highest granularity, potentially allowing fine-tuning of individual application components. However, accurately measuring or attributing energy consumption to specific processes is technically complex, often requiring kernel-level mechanisms like eBPF (used by *KubeWatt* [11]) or estimation models based on resource proxies like CPU usage (used by *Scaphandre* [14]).

Container-level: This level consolidates the consumption of all processes within a container, a key granularity for managing containerized applications. Tools typically achieve this by aggregating process-level estimates or by correlating container resource usage (via cgroups [14]) with node-level power data using attribution models (e.g., *Kepler*, *KubeWatt* [11]). This area is a primary focus of current research and tool development [11].

Pod-level: A pod represents one or more tightly coupled containers scheduled together on the same node [17]. Pod-level energy consumption is the sum of the consumption of its constituent containers. This level is useful for understanding the energy footprint of a deployable application unit within Kubernetes. Tools like *Kepler*, *Scaphandre*, and *CEEMS* provide metrics aggregated at this level [11].

Node-level: This refers to the overall energy use of a Kubernetes worker node (physical or virtual). Measuring at this level is often simpler, using hardware sensors via IPMI/Redfish or external devices like smart PDUs [11]. The main difficulty is assigning this total energy to the various workloads (pods, containers, system processes) on the node. Although crucial for container-level attribution models like *KubeWatt* [11], node-level data alone offers little direct information on specific application efficiency.

C. Persistent Challenges in Energy Measurement

Despite the progress made by tools like Kepler and *CEEMS*, several fundamental challenges persist across the field, limiting the accuracy and comparability of energy measurements in Kubernetes.

Hardware Heterogeneity: Production clusters are almost never homogeneous. They typically comprise a diverse set of nodes with varying generations of CPUs, GPUs, and other accelerators. Each piece of hardware has a distinct power profile, making it challenging to create a single, unified power model that is valid across the entire heterogeneous fleet. A model that has been trained on one server generation can be very inaccurate on another generation.

Measurement Overhead: Measurement itself consumes resources. Monitoring frameworks need CPU cycles, memory, and network bandwidth, each consuming its own share of energy. Even when tools such as *CEEMS* are frugal in their demands, with scrapes taking merely microseconds, this overhead can become a serious consideration in large clusters.

Lack of Standardization: The cloud-native platform today lacks a uniform set of measurements, APIs, and methodologies to measure and compare energy usage. In the absence of a common reference point, making like-for-like comparisons of the effectiveness of different tools, schedulers, and optimization methods is difficult. Observations reported in one research

work might turn out to be non-reproducible or non-comparable to another.

III. Energy-Aware Scheduling and Management Architectures

Building on energy monitoring insights, advanced tools and technologies are actively improving Kubernetes cluster energy efficiency. Moving beyond observation, these solutions employ intelligent controls for workload scheduling and energy resource management. Key developments include sophisticated energy-aware scheduling systems that optimize workload placement based on energy or carbon factors, alongside novel architectural patterns leveraging Kubernetes for enhanced energy management.

A. Energy-Aware Scheduling Systems

The standard Kubernetes scheduler determines pod placement on nodes, mainly considering resource availability (CPU, memory) and user constraints like affinity rules or taints/tolerations [18]. Extending this, energy-aware schedulers incorporate energy consumption or related metrics like carbon intensity into the decision process, forming a significant research area.

Several approaches exemplify this field. *CarbonScaler* dynamically adjusts server allocation for batch workloads based on fluctuating grid carbon intensity, using forecasting and an optimal greedy algorithm to achieve substantial carbon savings (e.g., up to 32.9% for a variant [20]) often without major delays [19], [20]. *Smart-Kube* uses Deep Reinforcement Learning (DRL) to automatically learn fair, energy-aware scheduling policies, adapting to cluster changes to manage resource utilization and energy use without extensive manual tuning [1]. The *Low Carbon Kubernetes Scheduler* prioritizes spatial shifting, assigning pods to nodes in geographic regions with the lowest current carbon intensity, effectively moving energy demand to cleaner locations [15]. Similarly, *GreenCourier* spatially shifts serverless functions [22].

Opportunistic scheduling using Graph Neural Networks (GNNs) models complex data center relationships to predict the energy impact of container placement, achieving energy reductions (e.g., 6.2% average) compared to default Kubernetes scheduling [23]. GNNs show promise for broader cloud resource management tasks [24], [25]. Another tactic is temporal shifting, delaying non-essential workloads to times with lower grid carbon intensity, potentially increasing job duration but reducing emissions [26], with architectures like *GAIA* aiming to balance carbon, performance, and cost [27].

The following table summarizes the core characteristics of these advanced scheduling systems:

TABLE II
Comparision of Energy-Aware Scheduling Systems

Scheduler Name	Core Algorithm / Technology	Primary Optimization Goal(s)	Key Strategy
CarbonScaler	Greedy Algorithm	Minimize Carbon Emissions	Dynamic resource allocation (”Carbon Scaling”)
Smart-Kube	Deep Reinforcement Learning	Minimize Energy, Maintain Pod Ratio	DRL-based node allocation/reallocation
Low Carbon Scheduler	Heuristic Ranking	Minimize Carbon Emissions	Spatial shifting of workloads to low-carbon regions
GNN-based Scheduler	Graph Neural Network (GNN)	Minimize Energy, Reduce SLA Violations	Opportunistic scheduling based on GNN power prediction

There is significant research interest in leveraging AI/ML, particularly Reinforcement Learning and GNNs, for Kubernetes scheduling [28], offering adaptability advantages over traditional heuristics [18] or optimization algorithms [19]. These AI/ML methods enable more autonomous and potentially effective energy-aware systems [1], [2]. The distinct carbon-aware strategies—temporal (when [26]) versus spatial (where [15])—highlight that the optimal approach depends on specific workload and infrastructure contexts.

B. Energy Management Architectures

Innovation is also occurring in the architectural patterns used for energy management, sometimes leveraging Kubernetes itself as a platform:

- **Kubernetes-Container-Cluster-Based Architecture for EMS:** This approach focuses on enhancing the reliability and resource utilization of Energy Management Systems (EMS) - the software systems used to monitor and control energy infrastructure (like power grids) - by deploying the EMS components themselves as containerized applications on a Kubernetes cluster [29]. By leveraging Kubernetes' features like container isolation, fault tolerance, and self-healing [29], this architecture aims to improve EMS uptime and resilience compared to traditional monolithic or Service-Oriented Architecture (SOA) deployments. A dynamic Pod fault-tolerant model, potentially based on Markov theory, can be used to automatically adjust the redundancy of EMS microservices to meet reliability targets while optimizing resource usage [29]. Studies suggest this approach can significantly reduce annual failure time and improve hardware utilization for the EMS itself [29].
- **Cloud-Based Smart Energy Framework:** Kubernetes is also being utilized as a scalable backend for data analytics in the broader smart energy domain [30]. Frameworks like the one developed for Chulalongkorn University's Building Energy Management System (CU-BEMS) use Kubernetes to orchestrate containerized data analytic workloads [30]. By employing a divide-and-conquer paradigm and parallel processing across Kubernetes pods, these frameworks can accelerate the analysis

of large datasets generated by smart meters and energy sensors [30]. Similarly, Kubernetes can host platforms for training and deploying ML models used in energy consumption prediction [31].

IV. Future Directions

After reviewing current monitoring tools, systems, and scheduling techniques, there are many strategies that can be applied by practitioners to make their Kubernetes deployments more energy efficient. However, substantial research challenges remain, highlighting promising avenues for future innovation.

A. Strategies for Sustainable Kubernetes Deployments

The insights gained from the examination of monitoring tools and advanced scheduling platforms can be synthesized in a collection of actionable tactics that organizations can leverage to ensure their Kubernetes deployments are more energy efficient. Tactics vary from fundamental configuration practice to the adoption of advanced, AI-driven tools, and provide an iterative path to sustainability. Table III - *(Comparison of different kubernetes deployment Strategy)* provides a convenient reference, pairing high-level tactics with the actual Kubernetes features and tools facilitating them.

B. Key Gaps and Remaining Challenges

Even though there has been progress, some major challenges still make it hard for energy efficiency measures to be widely used and fully effective in Kubernetes.

- **Lack of Standardized Measurement Metrics & Benchmarks:** As previously discussed, the absence of universally accepted metrics, measurement protocols, and benchmarks makes it difficult to objectively compare different tools, validate savings claims, and establish best practices [12].
- **Real-World Validation and Scalability:** Many research proposals are validated primarily through simulation or small-scale experiments. Demonstrating effectiveness and scalability in large, dynamic, real-world production environments remains a persistent challenge [1].

C. Promising Directions for Future Research and Development

Addressing the remaining challenges requires continued innovation across multiple fronts. Key future directions include:

- **AI-driven Optimization Opportunities:** Further research into applying AI/ML techniques for more accurate predictive modeling (of workload demand, energy consumption, carbon intensity), intelligent and adaptive resource allocation (scheduling, autoscaling), automated policy generation, and anomaly detection related to energy inefficiency [2]. This includes using AI not just to optimize Kubernetes, but also using AI for optimizing cloud-native systems more broadly [5].
- **Development of Multi-Objective Optimization Frameworks:** Creating more robust theoretical frameworks and

TABLE III
Comparison of different kubernetes deployment Strategies

Strategy	Description	Kubernetes Mechanisms/Tools	Key Benefit
Dynamic Resource Allocation	Continuously right-size pod resource requests and limits to align with actual workload usage, eliminating the waste caused by static over-provisioning.	Vertical Pod Autoscaler (VPA), Horizontal Pod Autoscaler (HPA), AI-powered optimization platforms.	Reduces idle reserved capacity on nodes, leading to more efficient bin-packing and higher overall utilization.
Workload Consolidation	Intelligently pack workloads onto the minimum number of nodes required to satisfy demand, allowing idle nodes to be deprovisioned and powered down.	Cluster Autoscaler, Karpenter , Pod Affinity/Anti-Affinity rules.	Maximizes energy savings by eliminating the significant static power draw of idle servers.
Adopt Energy-Aware Scheduling	Replace or extend the default Kubernetes scheduler with a custom scheduler that explicitly optimizes for energy consumption or carbon emissions.	Custom Scheduler Plugins (for tools like RLKube or CarbonScaler), Scheduler Extenders.32	Directly targets energy and carbon as primary optimization metrics, rather than treating them as byproducts of resource utilization.
Hardware-Aware Deployment	Ensure workloads are placed on the hardware best suited for them in terms of power efficiency, especially in heterogeneous clusters.	Node Labels, Node Feature Discovery (NFD), Topology Manager 38, GPU-specific schedulers like PWR.42	Leverages the inherent energy efficiency differences in diverse hardware to reduce overall consumption.
Workload Lifecycle Management	Automatically shut down non-production environments (e.g., development, staging, testing) during non-business hours to eliminate a major source of waste.	KubeGreen , custom Kubernetes CronJobs that scale deployments to zero replicas.	A simple yet highly effective strategy for eliminating the energy consumption of completely idle infrastructure.

practical tools (e.g., advanced schedulers, policy engines) that can explicitly model and systematically optimize for multiple, potentially conflicting, objectives simultaneously (energy, performance, cost, carbon, fairness, etc.) [20].

- **Standardized Benchmarking Methodologies:** Establishing "standard benchmarks", representative workloads, metrics, and evaluation methodologies specifically for assessing the energy efficiency and performance trade-offs of Kubernetes tools and strategies. This is crucial for enabling fair comparison, reproducibility, and driving progress in the field [9].

The parallel focus on edge computing introduces new challenges and opportunities, demanding specialized solutions that account for the unique constraints of distributed, resource-limited environments [32].

V. Conclusion

The optimization of energy-efficient Kubernetes operations is an imperative step towards a more sustainable cloud-native future. This survey reviewed the current state of energy efficiency in Kubernetes, a highly relevant topic due to the platform's widespread use in cloud computing. Accurate measurement and continuous monitoring play a vital role, forming the foundation for effective energy optimization efforts. Although significant advances have been made in developing energy visibility tools and smart workload placement algorithms, optimizing energy use in complex Kubernetes configurations remains an ongoing challenge critical to both environmental sustainability and cost savings.

Despite recent advancements, several key challenges persist, most notably the lack of standard measures for energy analysis, which does not allow comparison and verification of novel approaches. Furthermore, the need for research solutions to be adaptable to real-world production environments remains a key barrier. Breaking through these barriers is essential for large-scale implementation of energy conservation strategies. Future work is heading for more sophisticated, autonomous directions, including AI-driven predictive modeling, dynamic resource planning, and balanced systems that balance energy consumption with performance, cost, and emissions.

References

[1] S. Ghafouri, S. Abdipoor, and J. Doyle, "Smart-Kube: Energy-aware and fair Kubernetes job scheduler using deep reinforcement learning," in *2023 IEEE 8th International Conference on Smart Cloud (SmartCloud)*, 2023, pp. 154-163.

[2] R. C. Thota, "Optimizing Kubernetes workloads with AI-driven performance tuning in AWS EKS," *Int. J. Sci. Res. Arch.*, vol. 9, no. 2, pp. 1063-1073, 2023.

[3] CNCF, "Model, view, and reduce your workload carbon emission by Crane in a declarative way," *CNCF Blog*, 2023. [Online]. Available: https://www.cncf.io/blog/2023/03/27/model-view-and-reduce-your-workload-carbon-emission-by-crane-in-a-declarative-way

[4] C. Centofanti, J. Santos, V. Gudepu, and K. Kondepu, "Impact of power consumption in containerized clouds: A comprehensive analysis of open-source power measurement tools," Computer Networks, vol. 245, p. 110371, May 2024.

[5] CNCF TAG Runtime & WG Cloud Native AI, "CNCF cloud native AI white paper," *CNCF*, 2024. [Online]. Available: https://tag-runtime.cncf.io/wgs/cnaiwg/whitepapers/cloudnativeai/

[6] E. S. Mkoba and M. A. A. Saif, "A survey on energy efficient with task consolidation in the virtualized cloud computing environment," *International Journal of Research in Engineering and Technology (IJRET)*, vol. 3, no. 3, pp. 70-73, 2014.

[7] O. Şereflişan and M. Koyuncu, "A review study on energy consumption in cloud computing," *IJFMR - International Journal For Multidisciplinary Research*, vol. 6, no. 1, 2024.

[8] E. A. Santos, C. McLean, C. Solinas, and A. Hindle, "How does docker affect energy consumption? Evaluating workloads in and out of Docker containers," *Journal of Systems and Software*, vol. 146, pp. 14-25, 2018.

[9] B. Kocot, P. Czarnul, and J. Proficz, "Energy-aware scheduling for high-performance computing systems: A survey," *Energies*, vol. 16, no. 2, p. 890, 2023.

[10] E. Suarez et al., "Energy efficiency trends in HPC: what high-energy and astrophysicists need to know," 2025, arXiv:2503.17283. [Online]. Available: https://arxiv.org/abs/2503.17283

[11] B. Pijnacker, B. Setz, and V. Andrikopoulos, "Container-level energy observability in Kubernetes clusters," 2025, arXiv:2504.10702. [Online]. Available: https://arxiv.org/abs/2504.10702

[12] N. K. Shankaranarayanan et al., "POET: A platform for O-RAN energy efficiency testing," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, 2024, pp. 1-5.

[13] CNCF, "The road to Scaphandre v1.0: Challenges and improvements to come on IT energy consumption evaluation," *CNCF Blog*, 2023. [Online]. Available: https://www.cncf.io/blog/2023/10/11/the-road-to-scaphandre-v1-0-challenges-and-improvements-to-come-on-it-energy-consumption-evaluation/

[14] Hubblo, "Scaphandre documentation," [Online]. Available: https://hubblo-org.github.io/scaphandre-documentation/

[15] A. James and D. Schien, "A low carbon Kubernetes scheduler," in *ICT for Sustainability (ICT4S)*, 2019.

[16] M. Paipuri, "CEEMS: A resource manager agnostic energy and emissions monitoring stack," in *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2024, pp. 1862-1866.

[17] G. Turin, A. Borgarelli, S. Donetti, F. Damiani, E. B. Johnsen, and S. L. Tapia Tarifa, "Predicting resource consumption of Kubernetes container systems using resource models," *Journal of Systems and Software*, vol. 203, p. 111750, 2023.

[18] R. Furnadzhiev, M. Shopov, and N. Kakanakov, "Efficient orchestration of distributed workloads in multi-region Kubernetes cluster," *Computers*, vol. 14, no. 4, p. 114, 2025.

[19] W. A. Hanafy, Q. Liang, N. Bashir, D. Irwin, and P. Shenoy, "CarbonScaler: Leveraging cloud workload elasticity for optimizing carbon-efficiency," in *Abstracts of the 2024 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, 2024, pp. 49-50.

[20] A. Lechowicz, R. Shenoy, N. Bashir, M. Hajiesmaili, A. Wierman, and C. Delimitrou, "Carbon- and Precedence-Aware Scheduling for Data Processing Clusters," in Proceedings of the ACM SIGCOMM 2025 Conference, São Francisco Convent Coimbra Portugal: ACM, Sept. 2025, pp. 1241-1244.

[21] M. Amaral et al., "Kepler: A Framework to Calculate the Energy Consumption of Containerized Applications," 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), pp. 69-71, July 2023.

[22] M. Chadha, T. Subramanian, E. Arima, M. Gerndt, M. Schulz, and O. Abboud, "GreenCourier: Carbon-aware scheduling for serverless functions," in *Proceedings of the 9th International Workshop on Serverless Computing (WoSC '23)*, 2023, pp. 18-23.

[23] P. Raith et al., "Opportunistic energy-aware scheduling for container orchestration platforms using graph neural networks," in *2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, 2024, pp. 299-306.

[24] C. Meng, S. Song, H. Tong, M. Pan, and Y. Yu, "DeepScaler: Holistic autoscaling for microservices based on spatiotemporal GNN with adaptive graph learning," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2023, pp. 53-65.

[25] T. Theodoropoulos et al., "GNOSIS: Proactive Image Placement Using Graph Neural Networks & Deep Reinforcement Learning," in 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA: IEEE, July 2023, pp. 120-128.

[26] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, "Let's wait awhile: how temporal workload shifting can reduce carbon emissions in the cloud," in *Proceedings of the 22nd International Middleware Conference (Middleware '21)*, 2021, pp. 260-272.

[27] W. A. Hanafy, Q. Liang, N. Bashir, A. Souza, D. Irwin, and P. Shenoy, "Going green for less green: Optimizing the cost of reducing cloud carbon emissions," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS '24)*, 2024, pp. 479-496.

[28] Z. Xu, Y. Gong, Y. Zhou, Q. Bao, and W. Qian, "Enhancing Kubernetes automated scheduling with deep learning and reinforcement techniques for large-scale cloud computing optimization," in Ninth International Symposium on Advances in Electrical, Electronics, and Computer Engineering (ISAEECE 2024), P. Siano and W. Zhao, Eds., Changchun, China: SPIE, Oct. 2024, p. 175.

[29] Z. Li, H. Wei, Z. Lyu, and C. Lian, "Kubernetes-container-cluster-based architecture for an energy management system," *IEEE Access*, vol. 9, pp. 84596-84604, 2021.

[30] K. Saengkaenpetch and C. Aswakul, "Cloud-based smart energy framework for accelerated data analytics with parallel computing of orchestrated containers: Study case of CU-BEMS," in *Proceedings of the 3rd International Conference on Advanced Information Science and System (AISS 2021)*, 2021, pp. 1-6.

[31] P. Pääkkönen, D. Pakkala, J. Kiljander, and R. Sarala, "Architecture for enabling edge inference via model transfer from cloud domain in a Kubernetes environment," *Future Internet*, vol. 13, no. 1, p. 5, 2020.

[32] CNCF, "Five critical shifts for cloud native at a crossroads," *CNCF Blog*, 2025. [Online]. Available: https://www.cncf.io/blog/2025/04/14/five-critical-shifts-for-cloud-native-at-a-crossroads/