When Green Computing Meets Performance and Resilience SLOs

Haoran Qiu[1], Weichao Mao[1], Chen Wang[2], Saurabh Jha[2], Hubertus Franke[2], Chandra Narayanaswami[2],
Zbigniew Kalbarczyk[1], Tamer Başar[1], and Ravishankar Iyer[1]

[1]UIUC, [2]IBM Research

***Abstract*—This paper addresses the urgent need to transition to global net-zero carbon emissions by 2050 while retaining the ability to meet joint performance and resilience objectives. The focus is on the computing infrastructures, such as hyperscale cloud datacenters, that consume significant power, thus producing increasing amounts of carbon emissions. Our goal is to (1) optimize the usage of green energy sources (e.g., solar energy), which is desirable but expensive and relatively unstable, and (2) continuously reduce the use of fossil fuels, which have a lower cost but a significant negative societal impact. Meanwhile, cloud datacenters strive to meet their customers' requirements, e.g., service-level objectives (SLOs) in application latency or throughput, which are impacted by infrastructure resilience and availability. We propose a scalable formulation that combines sustainability, cloud resilience, and performance as a joint optimization problem with multiple interdependent objectives to address these issues holistically. Given the complexity and dynamicity of the problem, machine learning (ML) approaches, such as reinforcement learning, are essential for achieving continuous optimization. Our study highlights the challenges of green energy instability which necessitates innovative ML-centric solutions across heterogeneous infrastructures to manage the transition towards green computing. Underlying the ML-centric solutions must be methods to combine classic system resilience techniques with innovations in real-time ML resilience (not addressed heretofore). We believe that this approach will not only set a new direction in the resilient, SLO-driven adoption of green energy but also enable us to manage future sustainable systems in ways that were not possible before.**

***Index Terms*—sustainability, green energy, cloud computing, resilience, machine learning, machine learning resilience**

I. Introduction

Motivation. It has been reported that cloud datacenters' carbon emissions already contribute 2–3% of the overall global carbon footprint, and it has been estimated that they will account for 8% by 2030 [9]. Meanwhile, constantly evolving computing paradigms (e.g., microservices [17], [34], serverless computing [6], [16], and machine learning (ML) [8], [42]) are demanding increasing amounts of power. The energy issues are being further exacerbated by challenges in security and reliability (e.g., Spectre defenses [10]). Given that the underlying hardware technologies have reached a plateau as they approach the limits of their ability to scale with respect to performance and power usage effectiveness, achieving carbon efficiency for a sustainable future is a daunting challenge.

Challenges. As the use of green energy becomes more pervasive [2], [4], [18], increasing the adoption of green energy in cloud datacenters can scale down the carbon footprint. However, to achieve that, dependable delivery of customer-specific cloud operations (especially for critical societal applications, such as hospitals and transportation infrastructures) must be an integral part of future sustainable computing. The major challenges to achieving that goal are outlined below:

- **[C1]** *Fundamental Trade-off between Sustainability and Cloud SLOs.* Cloud datacenter operations have service-level objectives (SLOs) that detail performance and resilience requirements [15] regarding latency, throughput, and availability. Sustainable computing requires both sustainable energy costs (by minimizing the carbon footprint) and sustainable cloud operations (by meeting SLOs). Conversely, meeting stringent SLOs can incur high energy costs (e.g., due to overprovisioning). Cloud datacenters require careful design and optimization in dealing with this trade-off.
- **[C2]** *Disruption in Energy Optimization Due to Resilience Management.* Failure mitigation and service recovery protocols in cloud datacenters are developed to handle various hardware and software failures (e.g., network link failures and power outages) [19], [31], [32]. However, classic system resilience introduces disruptions to power optimization by incurring additional energy consumption (due to redundancy, migration, and checkpointing). In addition, as ML inference engines are increasingly integrated with today's cloud datacenters [7], classic system resilience does not take into account the impact of errors of ML inference, out-of-distribution situations, and data/model uncertainties. Co-designing power and resilience management is required to provide fast failure recovery and differential treatment to critical/non-critical services to minimize disruptions while optimizing carbon footprint.
- **[C3]** *Variability in Green Energy Supply and Dynamic Workload.* Green energy sources are inherently unstable [18], and cloud datacenter workloads also exhibit dynamically varying spatial and temporal patterns. Combined with **[C1]**, this requires a continuously optimized trade-off between cloud SLO violations and carbon emissions, posing a challenging multi-objective optimization problem.
- **[C4]** *Lack of an Application-aware Power Control Plane.* Substantial efforts have been made towards adopting a *top-down* approach in maximizing green energy usage, such as workload shifting either spatially or temporally based on predictions of carbon intensity [39]. However, conservative power control misses energy-saving opportunities, while

2833-292X/24/$31.00 ©2024 IEEE
DOI 10.1109/DSN-S60304.2024.00015

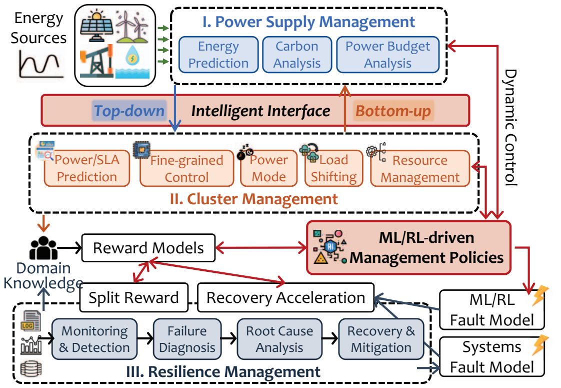

Fig. 1. Our contribution towards dependable green computing with an *intelligent interface* and a holistic framework of SLO-aware energy cost optimization, cluster management, and resilience management.

application-agnostic aggressive power control can lead to SLO violations [38]. Therefore, it is necessary to take a scalable, application-aware [3], *bottom-up* optimization approach and incorporate hardware/software co-design.

Our Approach. Achieving consistent service-level performance and resilience objectives must be an integral part of any assured green energy usage in cloud datacenters. We aim to reinvent cloud infrastructure with *SLO-aware energy efficiency* as the top priority. With a theoretical optimization formulation (§II), we address the problem from an ML perspective that has been shown to be successful in optimizing cloud efficiency [7], [26], [34], [37]. Fig. 1 presents an overview of our approach, which consists of three main novel components:

- ***An intelligent interface between power supply management and cluster management*** (§III-B) for joint optimization of the carbon footprint and cloud SLOs. The interface will enable both (1) top-down energy cost optimization, by enforcing the temporally varying power cap based on predicted carbon intensity, and (2) bottom-up SLO-aware power management (to address [C1] and [C3]), by predicting minimal power demand without SLO violations.
- ***Multi-tier ML in hierarchical decision-making*** (§III-A) for holistic, bottom-up datacenter power-resource management that is application-centric and can be executed efficiently at scale (to address [C4]). Conventional approaches are largely based on handcrafted heuristics that have become challenging to generate given the variations across heterogeneous cloud environments and workloads and rapid innovations across the system stack. We propose a hierarchical decision-making framework driven by (1) a multi-tier ML model to achieve combined intelligence in multi-objective optimization, and (2) leader-follower game formulation.
- ***Split reward models and failure recovery acceleration*** (§III-C) for SLO-aware energy optimization under datacenter failures (to address [C2]). Split reward functions allow the ML models to learn differential policies under various failure recovery procedures and for applications with diverse levels of criticality. The failure recovery acceleration module will coordinate cluster management and resilience management to minimize disruption to energy optimization. In addition, ***ML agent failures*** can be critical and inter-related with classic reliability and performance failures. ML agent resilience requires fast detection, handling, and retraining for unseen cases that become out-of-distribution compared to the data on which the agent has been trained.

Contributions. This paper presents multidisciplinary work that brings together power systems and cloud systems engineering to achieve progress towards dependable green computing. The proposed solution tackles the unique challenges of classic systems resilience and ML agent failures, as cloud systems increasingly integrate with ML solutions whose resilience is hard to verify because of issues such as data uncertainty in dynamic and heterogeneous cloud environments.

II. Problem Statement & Formulation

The key factors that compete to achieve dependable adoption of green energy in cloud datacenters are sustainability, resilience, and performance. They must be balanced while mitigating potential instability and costs associated with green energy, particularly in the event of cloud system or ML engine failures. The ultimate goal is to continuously reduce the carbon footprint while scaling infrastructure sustainability. Cloud datacenter workloads are typically categorized into latency-critical (LC) jobs and best-effort (BE) jobs [44]. LC jobs are typically associated with SLOs with respect to either latency or throughput. BE jobs typically do not have any SLOs, but their *daily* throughput should be maintained at a predefined level (or with some tolerable degradation) [39].

To facilitate the discussion of our proposed ideas and future challenges, we start by offering a problem statement with a mathematical formulation of strategic interactions between the power and cluster management agents.

- *Time Window.* We assume that the total period $[0, T]$ for power-resource management is partitioned into sub-periods, say $[t_k, t_{k+1})$, which could be one hour or a half-hour [1], and is referred to as "time interval t".
- *Power Supply.* We model each energy source as $e \in E$, e.g., fossil fuels, solar energy, and wind energy. The power supply of energy source e is then $p_e = P_e(t)$ for any time interval t, and its carbon intensity is $c_e = C_e(t)$. The total power supply to a datacenter is then $PS(t) = \sum_{e \in E} P_e(t)$, and the combined carbon intensity of the total supply is $CI(t) = \sum_{e \in E} C_e(t) \cdot P_e(t)/PS(t)$.
- *Datacenter Power Consumption.* We define the power consumption of a datacenter as $PC(t) = PC_{IT}(t) \cdot PUE$, where PUE is the ratio between the total facility energy and IT equipment energy. A datacenter typically has a constant PUE that is dependent on the power efficiency of the datacenter's operations [14]. In this paper, we assume that only $PC_{IT}(t)$ is under our control and that it depends on the scheduled workload at time t, the number of machines that are running, and the power mode or core frequency on each running machine. Therefore, $PC_{IT}(t) = \sum_{s \in S(t)} PC_{IT}(s, t)$, where server s is from the total running server set $S(t)$.

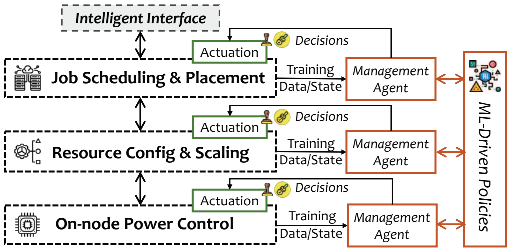

Fig. 2. Multi-tier ML in hierarchical decision-making.

- *Server Power Consumption.* The power consumption of a server $PC_{IT}(s,t)$ has been shown to be related to the processor (CPUs or accelerators, such as GPUs) utilization ($u(s,t)$) and running frequency ($f(s,t)$), and the relationship is called a *power profile* p [40]. Different processors can have different power profiles, and each power profile can be modeled as a power model F_p, typically parameterized by a neural network trained with profiling data. Therefore, the power consumption of a server can be defined as $PC_{IT}(s,t) = F_p(u(s,t), f(s,t))$.
- *Cluster Management Actions.* (1) Determine the number of servers in use, i.e., $S(t)$; (2) determine processor frequency on each server $f(s,t)$ by fine-grained core-level frequency tuning or server-level power capping; (3) schedules when jobs run or stop running (e.g., BE jobs can be delayed to run when carbon intensity is lower).
- *Constraints and Objectives.* The objective is to minimize the carbon footprint of the datacenter over any period of time T, i.e., to minimize $\sum_{t\ in[0,T]} PC(t) \cdot CI(t)$, constrained by the power cost budget, the SLOs of LC jobs, and the daily throughput degradation threshold for BE jobs.

III. Design Methods and Discussion

A. Multi-tier ML/RL Decision-Making and Control

In cluster management to serve datacenter workloads, as shown in Fig. 2, we divide the decisions into three interdependent layers: (1) job scheduling and placement, (2) resource allocation and scaling, and (3) on-node power control. A hierarchical set of decision-making actions can affect workload SLO preservation and power consumption. Starting from the interface (top), the set of jobs to run and to delay are determined by the job scheduling layer. Those jobs are then placed onto the set of running servers (i.e., $s(t)$) determined by the power control layer. The resource configuration and scaling layer allocates the resources to running jobs and dynamically scales the resource allocations at runtime. Collaboratively, the on-node power control layer adjusts control plane knobs to reduce power consumption while meeting SLOs.

How can we achieve multi-objective optimization in a competitive, hierarchical decision-making framework? The power supply's objective is to minimize power consumption and the carbon footprint, while datacenter applications' objective is to maximize performance and availability. Existing learning-based approaches such as FIRM [34] and SIMPPO [36] can help achieve latency-critical (LC) job SLOs with resource autoscaling but require coordination with other tiers of decision-making agents to (1) optimize power consumption with processor frequency scaling [46], and (2) optimize for datacenter carbon footprint minimization by leveraging the constrained flexibility of best-effort (BE) jobs [39]. The game-theoretical formulation requires a reward model design to reconcile meeting all application demands (LC job SLOs and BE job daily throughput) and scaling down carbon footprint. We plan to design a multi-agent framework that can efficiently explore and exploit optimal policies in the multi-objective hierarchical decision-making framework.

B. Intelligent Interface in Power-Cluster Management

As shown in Fig. 1, the interface between the power supply management and cluster management modules supports both top-down optimization (i.e., shaping power demand based on carbon intensity) and bottom-up optimization (i.e., shaping power supply based on SLO-aware power demand). The interface API communicates the power supply, carbon intensity, and power consumption (demand) at each time interval $[(PS(t), CI(t), PC(t))]_{t\in[0,T]}$. In the top-down optimization, $PS(t)$ and $CI(t)$ are determined based on predictions of the carbon intensity variation of each energy source $C_e(t)$ and then passed down to cluster management for temporal/spatial load shaping or resource reprovisioning. In the bottom-up approach, the power demand $PC(t)$ is determined based on predictions of the workload and *what-if* analysis of potential management decisions (i.e., scheduling, resource allocation, and on-node power control). Note that after the *what-if* analysis, the $PC(t)$ can be a range instead of a scalar. $PC(t)$ is then passed to the power supply module that controls the mix of energy sources exploited to minimize the carbon footprint within the energy cost budget. We need an *intelligent interface* to learn global optimality under uncertainty by reconciling datacenter workload power demand with multi-source green energy availability and balancing top-down and bottom-up optimizations.

How can a stochastic game-theoretical formulation provide an efficient model for optimal solutions at scale? The game-theoretical formulation (§II) naturally forms a hierarchical decision-making (leader-follower) structure where the "leader" can be the power supply module and the "follower" can be the cluster management module. It could potentially be formulated as a leader-follower Stackelberg game [5], [29], [30], as the leader determines and announces its strategy first by *anticipating* the followers' policies, and the followers determine their strategies as the best response to the leader's strategy. Given the stochasticity in both green energy generation and datacenter workloads, finding the optimal solutions efficiently can be challenging. It is important to be resilient to situations such as blackouts caused by extreme weather events, as datacenters have limited power reserves (e.g., batteries). We plan to focus on the design of the contracts between both parties by decoupling different layers in §III-A.

How can widely different decision-making time scales for power supply and cluster management be reconciled to achieve a holistic solution? Power supply and carbon intensity

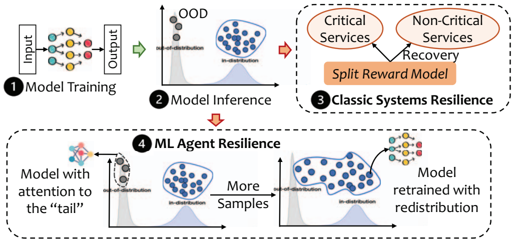

Fig. 3. System and ML agent resilience.

have more coarse-grained dynamics (e.g., on an hourly basis) than the minute/second level datacenter workload dynamics, so the decision-making frequency is different. High-frequency and low-frequency agents can be modeled as a hierarchical decision-making problem: low-frequency agents might adopt long-term learning strategies, while high-frequency ones might need to adapt quickly to immediate changes in the environment. To achieve effective decision-making, it is crucial to model the interactions, delays, and feedback loops accurately. Given the uncertainty and unpredictability of multi-source green energy availability, adapting solutions to changing conditions while optimizing multiple objectives in real-time adds another layer of complexity.

C. Split Reward Model for Systems-ML Resilience

Reward models or reward functions are commonly used to tune management policies in learning-based systems management tasks [34], [36], [46]. In power supply management, the reward is higher for a lower carbon footprint, and there is a penalty for higher-than-budget energy costs. Reward functions for cluster management aim to penalize low resource utilization and reward the meeting of LC job SLOs or BE job daily throughputs. However, cluster management policies learned under failure-free or normal operational conditions can fail or lead to sub-optimal decisions during datacenter failure recovery processes (e.g., because of networking failures or misconfigurations). For example, PARM [38] shows that outages or power-capping events can lead to severe performance degradation and agent policy failures. When cluster management agents are unaware of failure recovery procedures/strategies, agents' decisions can lead to cascading cluster outages or metastable failures [23], [38]. In addition, ML inference failures can be critical and interrelated with classic system failures. ML agent resilience requires fast detection, handling, and model retraining for tail cases that become out-of-distribution compared to the data on which the agent has been trained.

To address this gap, we propose *split reward models* that coordinate cluster management and resilience management. We introduce dedicated reward functions for failure recovery mode for keeping critical services running while attempting and accelerating system recovery.

How can we achieve fast, cloud service-aware failure recovery across power and cluster management? In terms of system failure recovery, the primary objective should be to restore system operation so as to ensure a successful application execution. Consequently, the power and cluster management software may either switch to a degraded mode or be disabled until the system fully recovers. In the presence of a failure, the system is already under significant stress, and all available (or operational) resources should be devoted to ensuring proper recovery. In addition, balancing the trade-offs between prioritizing critical services for faster recovery while also maintaining efficient for non-critical tasks requires careful design of the split models. We plan to incorporate *service-aware load control* to bridge the gap and coordinate cluster management with resilience management. A novel reward model for the recovery mode is needed to facilitate faster recovery (instead of only maintaining an optimal system state). Rapid identification of failure conditions, adaptation to energy availability, categorization of workloads, and appropriate reallocation of resources in real time pose significant challenges.

How can resilient ML agent performance be achieved if there are out-of-distribution or tail cases? As shown in Fig. 3, in addition to classic system resilience, ML agent resilience is also a challenge. At inference time, special attention should be given to tail cases that become out-of-distribution compared to the data on which the agent is trained [35]. ML agent resilience requires fast detection and handling (by retraining) of tail cases. Potential strategies include (1) falling back to heuristics-based approaches; (2) meta-learning tail samples to generate specialized models; and (3) re-distribution to merge specialized models into the original model.

D. Additional Discussion

We have not yet covered other challenges, such as multi-cluster and hardware heterogeneity.

Geographically Distributed Datacenters. The problem can be more complicated when considering multiple geographically distributed datacenters [1], [39], each of which can have a heterogeneous energy supply with different carbon intensity curves. Datacenter workloads could perhaps be migrated across clusters through leveraging their *spatial* flexibility.

Heterogeneous Hardware Accelerators. Heterogeneous hardware accelerators, especially those used for ML workloads (e.g., large model training and inference), are consuming more and more datacenter power. For instance, GPU devices in a cluster can be heterogeneous in terms of hardware, resource configurations (e.g., memory size), and power features [11], [25], [43], [47]. Device heterogeneity raises challenges in both job placement (e.g., which type of device to assign to a specific ML job) and power control (as the power efficiency differs across devices). We leave the study of this complicated optimization space to future work.

IV. Related Work

Datacenter Carbon Footprint Management. Substantial efforts have been made towards datacenter carbon footprint assessment [3] and reduction, mostly by adopting a top-down approach (e.g., workload shifting based on carbon intensity predictions) [1], [27], [39], [41]. For example, Carbon Explorer [1] takes datacenter power demand and renewable energy generation at specific geographic locations, and outputs load (power demand) distributions. However, these approaches ignore application intent (e.g., leading to performance degradation) and resilience requirements. Instead, this paper proposes a bottom-up approach for dependable green computing.

Datacenter Cluster Management. Cluster management decisions (e.g., resource allocation, job scheduling, and core frequency tuning) directly affect the datacenter power consumption and thus the carbon emissions [13], [21], [22], [28], [38], [49]. For example, CarbonScaler [22] greedily scales the resources allocated to applications in response to fluctuations in carbon intensity. ReTail [13] reduces the power consumption of latency-critical applications that have SLO constraints by predicting the minimum frequency based on a trained model. GreenDRL [49] uses an RL-based scheduler in a solar-energy-supported datacenter that minimizes energy costs.

Datacenter Resilience Management. Datacenter failure mitigation and recovery procedures have been developed for various causes (e.g., networking issues, power outages, and misconfiguration) that incur reduced computing capacity [12], [20], [24], [31], [33], [45], [48], [50]. However, without coordination with cluster management, the reduced capacity can lead to SLO degradation and low availability, while uninformed cluster management can incur metastable failures (i.e., a sustained effect of cascading or exacerbated failures) [23]. In addition, the integration of ML inference failures, data or model uncertainties, and runtime out-of-distribution errors is rarely addressed in the system's context.

References

[1] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. Carbon explorer: A holistic framework for designing carbon aware datacenters. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2023)*, ASPLOS 2023, page 118–132, New York, NY, USA, 2023. Association for Computing Machinery.

[2] Anup Agarwal, Jinghan Sun, Shadi Noghabi, Srinivasan Iyengar, Anirudh Badam, Ranveer Chandra, Srinivasan Seshan, and Shivkumar Kalyanaraman. Redesigning data centers for renewable energy. In *Proceedings of the 20th ACM Workshop on Hot Topics in Networks (HotNet 2021)*, pages 45–52, 2021.

[3] Rohan Arora, Umamaheswari Devi, Tamar Eilam, Aanchal Goyal, Chandra Narayanaswami, and Pritish Parida. Towards carbon footprint management in hybrid multicloud. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems (HotCarbon 2023)*, New York, NY, USA, 2023. Association for Computing Machinery.

[4] Shahzad Aslam, Sheraz Aslam, Herodotos Herodotou, Syed Muhammad Mohsin, and Khursheed Aurangzeb. Towards energy efficiency and power trading exploiting renewable energy in cloud data centers. In *Proceedings of the 2019 International Conference on Advances in the Emerging Computing Technologies (IEEE AECT 2019)*, pages 1–6. IEEE, 2020.

[5] Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*. SIAM Series in Classics in Applied Mathematics, Philadephia, USA, 1998.

[6] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, et al. *Serverless Computing: Current Trends and Open Problems*, pages 1–20. Springer Singapore, Singapore, 2017.

[7] Ricardo Bianchini, Marcus Fontoura, Eli Cortez, Anand Bonde, Alexandre Muzio, Ana-Maria Constantin, Thomas Moscibroda, Gabriel Magalhaes, Girish Bablani, and Mark Russinovich. Toward ML-centric cloud platforms. *Communications of the ACM*, 63(2):50–59, 2020.

[8] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[9] Zhiwei Cao, Xin Zhou, Han Hu, Zhi Wang, and Yonggang Wen. Toward a systematic survey for carbon neutral data centers. *IEEE Communications Surveys & Tutorials*, 24(2):895–936, 2022.

[10] Sunjay Cauligi, Craig Disselkoen, Daniel Moghimi, Gilles Barthe, and Deian Stefan. SoK: Practical foundations for software Spectre defenses. In *Proceedings of the 2022 IEEE Symposium on Security and Privacy (IEEE SP 2022)*, pages 666–680. IEEE, 2022.

[11] Shubham Chaudhary, Ramachandran Ramjee, Muthian Sivathanu, Nipun Kwatra, and Srinidhi Viswanatha. Balancing efficiency and fairness in heterogeneous GPU clusters for deep learning. In *Proceedings of the 15th European Conference on Computer Systems (EuroSys 2020)*, pages 1–16, 2020.

[12] Guo Chen, Yuanwei Lu, Yuan Meng, Bojie Li, Kun Tan, Dan Pei, Peng Cheng, Layong Luo, Yongqiang Xiong, Xiaoliang Wang, et al. FUSO: Fast multi-path loss recovery for data center networks. *IEEE/ACM Transactions on Networking*, 26(3):1376–1389, 2018.

[13] Shuang Chen, Angela Jin, Christina Delimitrou, and José F Martínez. ReTail: Opting for learning simplicity to enable QoS-aware power management in the cloud. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA 2022)*, pages 155–168. IEEE, 2022.

[14] Andrew A Chien, Chaojie Zhang, and Liuzixuan Lin. Beyond PUE: Flexible datacenters empowering the cloud to decarbonize. *USENIX HotCarbon 2022*, 2022.

[15] Jianru Ding, Ruiqi Cao, Indrajeet Saravanan, Nathaniel Morris, and Christopher Stewart. Characterizing service level objectives for cloud services: Realities and myths. In *Proceedings of the 2019 IEEE International Conference on Autonomic Computing (IEEE ICAC 2019)*, pages 200–206, 2019.

[16] Simon Eismann, Joel Scheuner, Erwin van Eyk, Maximilian Schwinger, Johannes Grohmann, Nikolas Herbst, Cristina L. Abad, and Alexandru Iosup. Serverless applications: Why, when, and how? *IEEE Software*, 38(1):32–39, 2021.

[17] Yu Gan, Yanqi Zhang, Dailun Cheng, Ankitha Shetty, Priyal Rathi, Nayan Katarki, Ariana Bruno, Justin Hu, Brian Ritchken, Brendon Jackson, et al. An open-source benchmark suite for microservices and their hardware-software implications for cloud & edge systems. In *Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2019)*, pages 3–18, 2019.

[18] Jiechao Gao, Haoyu Wang, and Haiying Shen. Smartly handling renewable energy instability in supporting a cloud datacenter. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS 2020)*, pages 769–778. IEEE, 2020.

[19] Peter Garraghan, Renyu Yang, Zhenyu Wen, Alexander Romanovsky, Jie Xu, Rajkumar Buyya, and Rajiv Ranjan. Emergent failures: Rethinking cloud reliability at scale. *IEEE Cloud Computing*, 5(5):12–21, 2018.

[20] Zhenyu Guo, Sean McDirmid, Mao Yang, Li Zhuang, Pu Zhang, Yingwei Luo, Tom Bergan, Madan Musuvathi, Zheng Zhang, and Lidong Zhou. Failure recovery: When the cure is worse than the disease. In *14th Workshop on Hot Topics in Operating Systems (HotOS XIV)*, 2013. https://www.usenix.org/conference/hotos13/session/guo.

[21] Kawsar Haghshenas, Somayyeh Taheri, Maziar Goudarzi, and Siamak Mohammadi. Infrastructure aware heterogeneous-workloads scheduling for data center energy cost minimization. *IEEE Transactions on Cloud Computing*, 10(2):972–983, 2020.

[22] Walid A. Hanafy, Qianlin Liang, Noman Bashir, David Irwin, and Prashant Shenoy. CarbonScaler: Leveraging cloud workload elasticity for optimizing carbon-efficiency. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(3), Dec 2023. https://doi.org/10.1145/3626788.

[23] Lexiang Huang, Matthew Magnusson, Abishek Bangalore Muralikrishna, Salman Estyak, Rebecca Isaacs, Abutalib Aghayev, Timothy Zhu, and Aleksey Charapko. Metastable failures in the wild. In *Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2022)*, pages 73–90, 2022.

[24] Karthick Jayaraman, Nikolaj Bjørner, Jitu Padhye, Amar Agrawal, Ashish Bhargava, Paul-Andre C Bissonnette, Shane Foster, Andrew Helwer, Mark Kasten, Ivan Lee, et al. Validating datacenters at scale. In *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM 2019)*, pages 200–213. 2019.

[25] Yimin Jiang, Yibo Zhu, Chang Lan, Bairen Yi, Yong Cui, and Chuanxiong Guo. A unified architecture for accelerating distributed DNN training in heterogeneous GPU/CPU clusters. In *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2020)*, pages 463–479, 2020.

[26] Ajaykrishna Karthikeyan, Nagarajan Natarajan, Gagan Somashekar, Lei Zhao, Ranjita Bhagwan, Rodrigo Fonseca, Tatiana Racheva, and Yogesh Bansal. SelfTune: Tuning cluster managers. In *Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2023)*, pages 1097–1114, 2023.

[27] Julia Lindberg, Yasmine Abdennadher, Jiaqi Chen, Bernard C Lesieutre, and Line Roald. A guide to reducing carbon emissions through data center geographical load shifting. In *Proceedings of the 12th ACM International Conference on Future Energy Systems*, pages 430–436, 2021.

[28] David Lo, Liqun Cheng, Rama Govindaraju, Luiz André Barroso, and Christos Kozyrakis. Towards energy proportionality for large-scale latency-critical workloads. *ACM SIGARCH Computer Architecture News*, 42(3):301–312, 2014.

[29] Sabita Maharjan, Quanyan Zhu, Yan Zhang, Stein Gjessing, and Tamer Başar. Dependable demand response management in the smart grid: A Stackelberg game approach. *IEEE Transactions on Smart Grid*, 4(1):120–132, 2013.

[30] Sabita Maharjan, Quanyan Zhu, Yan Zhang, Stein Gjessing, and Tamer Başar. Demand response management in the smart grid in a large population regime. *IEEE Transactions on Smart Grid*, 7(1):189–199, 2016.

[31] Justin Meza, Tianyin Xu, Kaushik Veeraraghavan, and Onur Mutlu. A large scale study of data center network reliability. In *Proceedings of the Internet Measurement Conference 2018*, pages 393–407, 2018.

[32] Mukosi Abraham Mukwevho and Turgay Celik. Toward a smart cloud: A review of fault-tolerance methods in cloud systems. *IEEE Transactions on Services Computing*, 14(2):589–605, 2018.

[33] Iyswarya Narayanan, Di Wang, Myeongjae Jeon, Bikash Sharma, Laura Caulfield, Anand Sivasubramaniam, Ben Cutler, Jie Liu, Badriddine Khessib, and Kushagra Vaid. SSD failures in datacenters: What? When? and Why? In *Proceedings of the 9th ACM International on Systems and Storage Conference*, pages 1–11, 2016.

[34] Haoran Qiu, Subho S. Banerjee, Saurabh Jha, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. FIRM: An intelligent fine-grained resource management framework for SLO-oriented microservices. In *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2020)*, pages 805–825, Berkeley, CA, USA, November 2020. USENIX Association.

[35] Haoran Qiu, Weichao Mao, Archit Patke, Shengkun Cui, Chen Wang, Hubertus Franke, Zbigniew T Kalbarczyk, Tamer Başar, and Ravishankar K Iyer. FLASH: Fast model adaptation in ML-centric cloud platforms. In *Proceedings of the 7th Annual Conference on Machine Learning and Systems (MLSys 2024)*, 2024.

[36] Haoran Qiu, Weichao Mao, Archit Patke, Chen Wang, Hubertus Franke, Zbigniew T. Kalbarczyk, Tamer Başar, and Ravishankar K. Iyer. SIMPPO: A scalable and incremental online learning framework for serverless resource management. In *Proceedings of the 13th Symposium on Cloud Computing (SoCC 2022)*, pages 306–322, New York, NY, USA, 2022. Association for Computing Machinery.

[37] Haoran Qiu, Weichao Mao, Chen Wang, Hubertus Franke, Alaa Youssef, Zbigniew T Kalbarczyk, Tamer Başar, and Ravishankar K Iyer. AWARE: Automate workload autoscaling with reinforcement learning in production cloud systems. In *Proceedings of the 2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 387–402, 2023.

[38] Haoran Qiu, Linghao Zhang, Hubertus Franke, Chen Wang, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. PARM: Adaptive resource allocation for datacenter power capping. In *Machine Learning for Systems Workshop at the Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023. https://nips.cc/virtual/2023/84453.

[39] Ana Radovanović, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, et al. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems*, 38(2):1270–1280, 2022.

[40] Suzanne Rivoire, Parthasarathy Ranganathan, and Christos Kozyrakis. A comparison of high-level full-system power models. *USENIX HotPower 2008*, 8(2):32–39, 2008.

[41] Haiying Shen, Haoyu Wang, Jiechao Gao, and Rajkumar Buyya. An instability-resilient renewable energy allocation system for a cloud datacenter. *IEEE Transactions on Parallel and Distributed Systems*, 34(3):1020–1034, 2023.

[42] Pramila P. Shinde and Seema Shah. A review of machine learning and deep learning applications. In *Proceedings of the 4th International Conference on Computing Communication Control and Automation (IEEE ICCUBEA 2018)*, pages 1–6. IEEE, 2018.

[43] Fengguang Song and Jack Dongarra. A scalable framework for heterogeneous GPU-based clusters. In *Proceedings of the 24th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2012)*, pages 91–100, 2012.

[44] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. Large-scale cluster management at Google with Borg. In *Proceedings of the 10th European Conference on Computer Systems (EuroSys 2015)*, Bordeaux, France, 2015. https://doi.org/10.1145/2741948.2741964.

[45] Guosai Wang, Lifei Zhang, and Wei Xu. What can we learn from four years of data center hardware failures? In *Proceedings of the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2017)*, pages 25–36. IEEE, 2017.

[46] Yawen Wang, Daniel Crankshaw, Neeraja J. Yadwadkar, Daniel Berger, Christos Kozyrakis, and Ricardo Bianchini. SOL: Safe on-node learning in cloud platforms. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2022)*, pages 622–634, New York, NY, USA, 2022. Association for Computing Machinery.

[47] Qizhen Weng, Wencong Xiao, Yinghao Yu, Wei Wang, Cheng Wang, Jian He, Yong Li, Liping Zhang, Wei Lin, and Yu Ding. MLaaS in the wild: Workload analysis and scheduling in large-scale heterogeneous GPU clusters. In *Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2022)*, pages 945–960, 2022.

[48] Xin Wu, Daniel Turner, Chao-Chih Chen, David A. Maltz, Xiaowei Yang, Lihua Yuan, and Ming Zhang. NetPilot: Automating datacenter network failure mitigation. In *Proceedings of the 2012 ACM SIGCOMM*, pages 419–430, 2012.

[49] Kuo Zhang, Peijian Wang, Ning Gu, and Thu D. Nguyen. GreenDRL: Managing green datacenters using deep reinforcement learning. In *Proceedings of the 13th Symposium on Cloud Computing (SoCC 2022)*, page 445–460, New York, NY, USA, 2022. Association for Computing Machinery.

[50] Shenglin Zhang, Ying Liu, Weibin Meng, Zhiling Luo, Jiahao Bu, Sen Yang, Peixian Liang, Dan Pei, Jun Xu, Yuzhi Zhang, et al. Prefix: Switch failure prediction in datacenter networks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):1–29, 2018.