Reinforcement Learning for Real-time Data Center Resource Optimization

Sarah Ahmed sa2436 Yuval Steimberg ys2335 Mitchell Krieger mak483

I. Abstract

The increasing demand for high-performance data centers in the digital economy necessitates efficient resource allocation strategies. Traditional static provisioning methods fall short in adapting to dynamic workloads, prompting the exploration of machine learning (ML) solutions. This paper investigates how reinforcement learning (RL), a subset of ML, can optimize real-time resource allocation in data centers. By dynamically adjusting resource configurations based on workload patterns, RL enhances performance, energy efficiency, and scalability. Case studies demonstrate RL's transformative potential in areas such as traffic optimization and predictive maintenance, laying a foundation for sustainable data center management.

II. The Evolving Landscape of Digital Infrastructure

As data centers evolve to meet the demands of the digital economy, their management becomes increasingly complex. The scale and diversity of workloads, coupled with the need for continuous availability and reliability, require intelligent systems to optimize resource allocation in real-time. Traditional resource management methods, such as static provisioning using machine learning models, often fail to adapt to shifting workload patterns and environmental conditions. To address these challenges, advanced machine learning techniques, particularly reinforcement learning (RL), have emerged as game-changing solutions. Unlike static methods, RL systems learn from environmental interactions to develop adaptive policies that optimize performance, reduce energy consumption, and enhance scalability. This adaptability ensures system resilience even under varying workloads (Thein, 2020). A key application of RL in data centers is real-time resource allocation. RL agents learn effective strategies—such as adjusting CPU frequencies, managing memory, and allocating virtual machines—based on workload demands. These strategies are dynamically refined over time, enabling autonomous optimization of resources without human intervention. By contrast, static methods cannot keep pace with rapidly changing conditions. The dynamic adaptability of RL enhances scalability and flexibility, which are essential for managing the increasing complexity of modern workloads (Thein, 2020). Machine learning models also excel in predictive maintenance, another critical aspect of data center management. Predictive models analyze historical and real-time data to identify potential equipment failures, enabling proactive intervention. This approach minimizes

downtime, extends equipment lifespan, and reduces maintenance costs. For instance, sensor data can predict failures in cooling systems or power supplies, ensuring uninterrupted operations of critical services (Gao, 2011). Energy efficiency is another priority for data center operators, given rising energy costs and sustainability demands. While traditional machine learning models optimize energy consumption by dynamically adjusting cooling systems, server utilization, and power states in response to workload patterns, RL further enhances these efforts. By incorporating energy efficiency into reward functions, RL agents strike a balance between performance and reduced power consumption. This dual focus lowers operational costs and supports sustainability goals, addressing global environmental concerns (Thein, 2020). Beyond resource allocation and maintenance, machine learning automates decision-making processes such as load balancing, network traffic management, and security. By analyzing real-time traffic, these systems dynamically adjust routing and prioritize critical operations, ensuring efficient service delivery. They also enhance security by detecting patterns indicative of threats, enabling rapid responses to potential breaches. Automation reduces manual intervention, freeing operators to focus on strategic tasks while maintaining operational efficiency (Gao, 2011). To evaluate data center efficiency and sustainability, Key Performance Indicators (KPIs) such as Power Usage Effectiveness (PUE), Data Center Infrastructure Efficiency (DCiE), Carbon Usage Effectiveness (CUE), and Water Usage Effectiveness (WUE) are essential. These metrics promote sustainable practices and efficient resource allocation. Operational metrics, including Server Utilization and Network Latency, ensure optimal performance, while high Availability/Uptime guarantees uninterrupted service delivery (Koomey, 2011; Gao, 2011).

III. Performance Tooling and Machine Learning in Data Centers

Machine learning addresses the nonlinear interdependencies in data centers, offering transformative solutions for optimization. By analyzing real-time data, it dynamically adjusts parameters like temperature and load distribution, enhancing system performance and energy efficiency. Predictive maintenance detects early signs of equipment failure, while RL optimizes compute, storage, and networking resources. These capabilities are vital for managing dynamic Information Technology loads, fluctuating environmental conditions, and evolving operational constraints (Jim Gao, 2014). Among these capabilities, efficient resource allocation stands out as a critical application where RL's dynamic adaptability truly shines. By overcoming the limitations of static approaches, RL enables real-time optimization of hardware configurations. For example, Microsoft's project Tuning Data Center Performance with Machine Learning by M. Gottscho et al. highlights how RL continuously refines resource management strategies, maximizing performance, minimizing energy use, and reducing operational costs. As the digital landscape expands, adopting machine

learning and RL technologies is no longer optional—it is essential for maintaining efficiency, sustainability, and competitiveness in modern data centers. M. Gottscho et al. research focuses on improving data center performance by leveraging machine learning techniques to optimize hardware, and firmware, configurations in real time. The primary goal is to balance energy efficiency and performance by adjusting settings such as CPU frequencies, DRAM timings, and cache configurations. Data centers host diverse workloads with unique performance requirements, and traditional static approaches often fail to adapt to their dynamic nature. Machine learning models in this project address these inefficiencies by sensing hardware performance metrics and optimizing configurations in real time. A key component of the M. Gottscho et al. research is the development of X-mem, a tool designed for detailed characterization of DRAM performance. DRAM, as a critical component of data center operations, impacts memory bandwidth, latency, and overall system throughput. Existing tools for measuring DRAM performance often lack granularity, prompting the creation of X-mem, which provides detailed insights into memory hierarchy and performance metrics such as throughput, latency, and power consumption. Its modular and extensible design allows compatibility with various hardware platforms, such as Intel Atom micro-servers and ARM-based boards. Using X-mem, researchers could analyze the impact of Non-Uniform Memory Access (NUMA) settings and cache configurations on system performance. For instance, they identified significant asymmetries in main memory performance caused by interactions between NUMA settings and page sizes. These insights allowed fine-tuning of cache and memory configurations to optimize both energy efficiency and performance. Furthermore, X-mem enabled the analysis of DRAM timing and frequency settings, revealing trade-offs between throughput and power consumption under different workloads. This granular understanding allowed for the dynamic adjustment of DRAM parameters to match real-time workload demands. Armed with data from tools like X-mem, M. Gottscho et al. developed machine-learning models for dynamic hardware optimization, an approach termed soft heterogeneity. Instead of relying on static configurations, this method adjusts hardware settings in response to workload characteristics. The core methodology, FXplore, combines machine learning models with graph algorithms to explore optimal firmware configurations, including hardware prefetching, CPU turbo boosting, DRAM turbo boosting, and hyperthreading. By training these models on historical workload data, FXplore predicts and assigns optimal configurations dynamically, significantly improving average runtime and energy efficiency. Despite these advancements, challenges in real-time resource allocation persist. Data centers host a wide variety of applications with diverse priorities and resource needs. Variability in workloads complicates the efficient allocation of resources, as peak traffic demands alternate with idle periods. Additionally, the complex dependen-

cies between hardware configurations and workload performance make manual tuning impractical. Optimizing parameters such as DRAM timings and cache settings requires detailed insights and robust methodologies like those demonstrated by X-mem and FXplore. Therefore, M. Gottscho, et al takes one step further to enhance the transformative potential of machine learning in data centers. By integrating advanced tools and more interactive Machine Learning models, data centers can achieve significant improvements in energy efficiency and system performance. However, the ongoing challenges of workload variability and complex interdependencies underscore the need for continuous innovation in real-time resource allocation. The development of dynamic, adaptive systems remains essential for the sustainable growth of data centers in the digital era. Therefore, advanced approaches like reinforcement learning (RL) for resource allocation needs become even more evident when considering dynamic and adaptive systems. RL offers a powerful framework for addressing the challenges of workload variability and complex interdependencies by enabling systems to learn optimal strategies through trial and error. In this context, RL leverages the interaction between an agent (e.g., the data center management system) and its environment (the data center infrastructure) to refine actions such as adjusting CPU frequencies or allocating memory, guided by feedback in the form of rewards like improved performance or reduced energy consumption. The essential components of RL include the agent, environment, state space, action space, and reward function. The RL agent is responsible for decision-making, learning policies that map system states, such as workload characteristics, to specific actions like optimizing hardware configurations. The environment, consisting of the data center's hardware and workloads, provides feedback based on the agent's decisions. The state space captures information such as CPU utilization, memory usage, network bandwidth, workload characteristics, and temperature. Meanwhile, the action space encompasses possible adjustments, including modifying memory settings, allocating or deallocating virtual machines, or managing power states. The reward function defines the system's objectives, often combining performance metrics like throughput and latency with considerations for energy efficiency. The RL agent learns optimal policies by exploring different actions and analyzing their impact on system performance and energy usage. Techniques like Q-learning, which estimates rewards for specific state-action pairs, and policy gradient methods, which directly map states to actions, are commonly used to develop strategies that balance competing objectives like maximizing performance while minimizing power consumption. M. Gottscho et al's exploration of machine learning for optimizing data center performance provides an excellent foundation for RL applications. Their research demonstrated how machine learning models could map workload characteristics to hardware configurations, such as DRAM timing and CPU turbo boosting. Because

of this dynamic adaptability, reinforcement learning extends beyond traditional X-mem tool methods by enabling continuous adaptation to real-time workload changes. For instance, an RL agent can monitor varying workloads and dynamically adjust configurations based on learned policies. During low-traffic periods, the agent could experiment with different settings to improve future strategies while maintaining stable performance. Additionally, by incorporating energy consumption into the reward function, the RL agent can optimize power usage during idle periods and maintain high performance during peak demand. Integrating reinforcement learning into data center management offers numerous benefits. It enables adaptability by continuously adjusting strategies to dynamic workloads and environmental conditions. RL systems also scale well, managing complex, large-scale environments with evolving workloads and hardware configurations. Moreover, RL automates decision-making processes, reducing manual intervention and operational overhead. By continuously learning and adapting, RL agents offer the potential to balance performance with energy efficiency, contributing to the sustainable growth of data centers.

IV. Recent Innovations using Reinforcement Learning in Data Centers

As the demand for data centers continues to grow, traffic optimization within data centers is still a critical optimization problem. There are three critical tasks in traffic optimization, congestion control, load balancing and flow scheduling. Historically, traffic optimization methods used hand crafted heuristics to measure and maximize performance in all three tasks. However, measuring and monitoring these heuristics can be time consuming, costly and often do not find an optimal configuration. This is because they often require manual insights from experts with application specific knowledge and statistics that are collected over long periods of time. In recent years, machine learning has been leveraged to help solve this problem. More specifically, Salman et al. (2017) argued that despite optimization problems in datacenters having diverse goals, they actually are similar in design and structure. Because they share these attributes and modern data center traffic is highly predictable, many problems could benefit from optimization techniques that generalize across different goals. This makes data center optimization ripe for machine learning in a large spectrum of problems. Salman et al. envisioned a framework, DeepConf, for using machine learning in data centers that leverages deep learning to create an intermediate representation of the state of the data center. This representation could then be fed to a reinforcement learning model which could be configured to tackle a variety of problems in the data center. Reinforcement learning works well in these data center problems because they are controlled environments with predefined rules (often can be formulated as Markov Decision Processes). In addition, reinforcement learning can adapt to changes in the environment and make adjustments to its decisions, which is ideal for traffic coming into data centers which can fluctuate.

Chen et al. (2018) attempted to answer if deep reinforcement learning could be applied at datacenter-scale to flow completion time. They note that a key problem of deep reinforcement learning is that there is a long latency between the collection of information from the environment and the production of actions (10s - 100s of milliseconds) when latency in data centers often needs to be submillisecond. However, flows can come in a variety of sizes, so flows of short sizes need faster decisions than flows of long sizes. In addition, most data center flows are shorter, but most traffic comes from long flows. To solve this problem, Chen et. al. proposed AuTO, a two level system that takes inspiration from the peripheral and central nervous systems in animals. AuTO attempts to minimize flow completion time in data centers while trying to ensure fairness amongst flows of different sizes and urges. AuTO's Peripheral System makes decisions locally for short flows on end hosts in the data center with minimal delay and records information about flows. The Central System on the other hand, makes decisions for longer flows that can handle delays and observe global performance for policy optimization. The Peripheral System's decisions are influenced then by the Central System's observations. By using this two-leveled system we split the work of traffic management into the fast decision making of the Peripheral System and the strategic optimizations that can be made by the Central System.

The Peripheral System has two key components, the enforcement module and the monitoring module. The enforcement module is responsible for the execution of the order of flows via a multi-level feedback queuing (MLFQ) algorithm. In MLFQ, there is a hierarchical set of queues that order flows for execution. All flows begin in the top queue and as data arrives are demoted to lower queues each time their length exceeds a certain threshold. The enforcement module only needs the local information about the flow length to make decisions about when to execute flows. The longest flows in the bottom queue are sent to the Central System by the monitoring module for decisions to be made by the Central System using global information. In addition, the monitoring module records flow sizes, completion times and other useful information to be sent to the Central System. This information acts as the environment for reinforcement learning.

The Central System is composed of two deep reinforcement learning agents: Short Flow Reinforcement Learning Agent (sRLA) and the long flow reinforcement learning agent (lRLA). The sRLA is responsible for optimizing the MLFQ thresholds in the Peripheral System. It is trained using the deep deterministic policy gradient algorithm which attempts to maximize the reward of a given action (adjusting the values of the thresholds up or down) given the environment data provided by the Peripheral System. The reward in this case is the ratio between flow completion time at the current time step compared to the previous. In other words, its goal is to minimize flow completion time by ensuring that the MLFQ thresholds are optimal. The

lRLA is responsible for making decisions about the rates, routes and priority for the long flows using the deep policy gradient algorithm. lRLA attempts to maximize the throughput of long flows.

Chen et al. were able to demonstrate that AuTO reduced traffic optimization decisions from weeks of manual heuristic study and investigation to milliseconds. In addition, it also achieved superior performance, providing up to 48% decrease in flow completion time. This demonstrates that reinforcement learning has the potential to not only automate the process of traffic optimization but also perform better than manual heuristics could in the first place.

Another area of importance in data centers is data storage and access. Customers rely on data centers for fast data access, so latency is a crucial metric to track. In order to minimize latency, we need to intelligently choose where in the data center to place data. One challenge with the traditional heuristics and models designed to tackle this problem in the past is that they were static and couldn't adapt to the dynamic workloads of a data center. Liao et al. (2020) introduced Low latency and Fast Convergence Data Storage (FLDS) to address this problem. LFDS uses Q-learning to assess the environment and make decisions about what node in the datacenter's network to store data on. Q-learning works by attempting to learn a bellman function Q that predicts which action in an environment will maximize a reward function that considers both a current reward and the possible value of a future reward. This information is stored in a Q-table which lists the set of possible environments and the possible rewards given a particular action. In FLDS, the reward for a given action (where to place and environment is the reciprocal of the weighted sum of read and write delays.

However, Liao et al. noted two problems with using Q-learning for data storage in data centers. First, is that because there are a vast number of possible states and actions in data centers the Q-table is huge and especially complex. This can make it difficult to generate actions in an efficient manner. Second, finding the recommended learning and discount rates for the reward function required for Q-learning to converge can be quite challenging. To account for these challenges, FLDS leverages a sparse input matrix method. Liao et al. empirically showed that Zipf's law applies to data access because only a few reads and writes appear frequently. This means that we can greatly reduce the size of the Q-table by only considering the few most frequent requests, without actually losing too much information about which action would yield the best result for the given environment. This sparse matrix is fed into a neural network so that it can make faster decisions. In addition, they also performed parameter optimization to make sure that this approach converges quickly. Ultimately, this method was able to decrease data access latency by 23.5% and increase convergence speed by 15%.

V. Conclusion

In conclusion, Reinforcement learning (RL) offers significant potential for opti-

mizing resource allocation in data centers by adapting to dynamic workload patterns in real-time. As RL agents continuously learn from their interactions with the environment, they can develop strategies that effectively balance competing objectives such as maximizing performance and minimizing energy consumption. The promising results from both traffic optimization and data storage demonstrate that reinforcement learning has the potential to address multiple aspects of data center management. By offering dynamic, adaptive solutions to traditionally static processes, RL can significantly improve operational efficiency, reduce latency, and optimize resource usage. These advancements highlight the transformative power of RL in the evolving landscape of data center operations. As cloud computing continues to drive demand for more scalable, energy-efficient solutions, integrating RL into data center management systems presents a critical opportunity to reduce operational costs, increase performance, and meet the growing needs of the digital economy.

REFERENCES

[1] T. Thein, M. M. Myo, S. Parvin, and A. Gawanmeh, "Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 10, 2020.

[2] J. Koomey, *Growth in Data Center Electricity Use 2005 to 2010*, Analytics Press, Oakland, CA, 2011.

[3] J. Gao and R. Jamidar, "Machine learning applications for data center optimization," Google White Paper, vol. 21, 2014.

[4] M. Dorigo and T. Stützle, "Ant Colony Optimization: Overview and Recent Advances," *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1689–1700, Sept. 2002, doi: 10.1109/JPROC.2002.801938.

[5] M. Gottscho, S. Govindan, B. Sharma, M. Shoaib, and P. Gupta, "X-Mem: A cross-platform and extensible memory characterization tool for the cloud," *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Uppsala, Sweden, 2016, pp. 263–273, doi: 10.1109/ISPASS.2016.7482101.

[6] S. Salman, C. Streiffer, H. Chen, T. Benson, and A. Kadav, "DeepConf: Automating Data Center Network Topologies Management with Machine Learning," in *Proceedings of the 2018 Workshop on Network Meets AI & ML (NetAI'18)*, ACM, New York, NY, USA, pp. 8–14, 2018. doi: 10.1145/3229543.3229554.

[7] L. Chen, J. Lingys, K. Chen, and F. Liu, "AuTO: scaling deep reinforcement learning for datacenter-scale automatic traffic optimization," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*, ACM, New York, NY, USA, pp. 191–205, 2018, doi: 10.1145/3230543.3230551.

[8] Z. Liao, J. Peng, Y. Chen, J. Zhang, and J. Wang, "A Fast Q-Learning Based Data Storage Optimization for Low Latency in Data Center Networks," *IEEE Access*, vol. 8, pp. 90630–90639, 2020, doi: 10.1109/ACCESS.2020.2994328.