Ubiquitous Technology Journal
Vol. 1, No. 1, February 2025, pp. 40~47
ISSN: 3079-5273, DOI:10.71346/utj.v1i1.11

Real-Time Workload Prediction and Resource Optimization for Parallel Heterogeneous High-Performance Computing Systems Architectures

Ayesha Aslam [1], Zhumakhanova Darya Anuarovna [2]
[1] School of Information Engineering, Chang'an University, Xi'an, China
[2] Physics Computer Science Department, Physics - Matematika, Shakarim University, Semey, Kazakhstan

Article Info

Article history:

Received December 24, 2024
Revised January 04, 2025
Accepted January 05, 2025

Keywords:

Adaptive resource management
heterogeneous parallel architectures
task scheduling; machine learning-driven optimization
Source allocation
data placement
energy efficiency
fault tolerance
workload prediction
high-performance computing.

ABSTRACT

The rapid advancements in heterogeneous parallel architectures, consisting of CPUs, GPUs, FPGAs, have introduced significant challenges in efficient resource management for high-performance computing systems. Static and heuristic-based approaches fail to address the adaptability required for handling varying workloads and hardware configurations which results in suboptimal performance and energy inefficiency. This research proposes a machine learning-driven adaptive resource management framework that dynamically optimizes task scheduling, resource allocation, and data placement. The framework employs regression models & reinforcement learning algorithms to predict workload behaviors, resource utilization, and task execution times in real time. Experimental results on heterogeneous testbed demonstrate a 21% reduction in task execution time, 18% improvement in energy efficiency, and 38% decrease in fault recovery time compared to conventional methods. These findings highlight the framework's ability to improve resource utilization while maintaining reliability and minimizing energy overhead. The work advances the field by introducing a unified approach that integrates machine learning for runtime optimization across heterogeneous systems. Practical implications include its applicability to large-scale scientific simulations and deep learning tasks, where adaptive resource management is critical. Future study can focus on enhancing prediction accuracy by advanced deep learning techniques and extending the framework to handle emerging hardware accelerators and edge computing environments.

Corresponding Author:

Ayesha Aslam
School of Information Engineering
Chang'an University

710000, Xi'an, Shaanxi, China
Email: 2022024905@chd.edu.cn

1. INTRODUCTION

In recent years, the advent of heterogeneous parallel architectures has revolutionized the domain of high-performance computing (HPC) by enabling the integration of diverse processing units such as CPUs, GPUs, and FPGAs within a single computing platform [1]. The growth in computational complexity across various scientific and industrial applications has necessitated this evolution, as conventional homogeneous architectures struggle to meet modern computational and energy efficiency demands [2]. Heterogeneous

systems offer immense potential to deliver unparalleled computational throughput, yet their non-uniform architecture introduces significant challenges for efficient resource management. These challenges stem from differences in compute capabilities, communication bandwidths, and energy consumption across processing units, complicating task scheduling and resource allocation [3]. The proliferation of complex workloads, particularly those involving data-driven tasks and real-time applications, further exacerbates the resource management challenges in heterogeneous systems [4]. Workloads exhibit dynamic characteristics, including variability in task size, execution time, and computational dependencies. Static scheduling and allocation approaches, traditionally designed for homogeneous systems, prove inefficient and often result in underutilization of resources, reduced performance, and excessive energy consumption [5]. Existing methods fail to adapt to runtime changes in workloads and hardware conditions, limiting the performance portability and scalability of applications deployed on heterogeneous platforms.

The integration of machine learning (ML)-driven techniques offers a promising pathway to address these issues by enabling intelligent decision-making for resource management at runtime. Machine learning models can analyze real-time execution data to predict workload behaviors, optimize resource allocation, and dynamically adjust task placement across heterogeneous hardware components [6-7]. Such methods introduce adaptive capabilities that allow systems to respond effectively to varying computational demands and hardware availability. By learning system behavior over time, these approaches achieve higher resource utilization, reduced task execution latencies, and improved energy efficiency.

Energy efficiency and fault tolerance are paramount considerations in modern heterogeneous systems, as large-scale parallel computing platforms often operate under strict power budgets and are susceptible to hardware failures [8]. Resource management strategies must consider these aspects by minimizing power consumption without compromising performance and introducing mechanisms to detect and recover from faults during execution. Current solutions offer partial improvements, but they often remain disconnected from real-time execution environments and fail to provide holistic optimization for task scheduling, data placement, and energy management in heterogeneous systems [9].

This study addresses the aforementioned challenges by proposing an adaptive resource management framework that dynamically optimizes task scheduling, resource allocation, and data placement using machine learning-driven approaches. The framework operates in real time, enabling it to adapt to workload variability and hardware heterogeneity while balancing performance, energy efficiency, and fault tolerance. The novelty of this work lies in its ability to utilize predictive ML models for runtime optimization, an aspect often overlooked in existing methods, which primarily focus on static or heuristics-based solutions. The proposed techniques are evaluated on real-world heterogeneous platforms to ensure practical feasibility and demonstrate their effectiveness in addressing the complex demands of modern parallel computing environments.

The contributions of this work include the development of machine learning models for predicting workload behaviors and resource utilization patterns in heterogeneous systems. These models are integrated into an adaptive scheduling mechanism that dynamically assigns tasks to processing units based on predicted performance and energy metrics. A data placement strategy is introduced to minimize memory access bottlenecks and ensure efficient communication between tasks operating on different hardware units. Likewise, the study incorporates fault tolerance mechanisms that detect hardware failures and reallocate resources to ensure uninterrupted execution. The proposed techniques are implemented and rigorously evaluated on real-world platforms comprising CPUs, GPUs, and FPGAs, highlighting their advantages in terms of performance, scalability, and energy efficiency.

By addressing the challenges of resource management in heterogeneous parallel architectures, this study contributes to bridging the gap between theoretical advancements and practical implementations. It offers a systematic approach for developing intelligent, adaptive, and fault-tolerant resource management strategies tailored to the needs of modern HPC systems. The experimental results demonstrate significant performance improvements compared to existing methods, underscoring the potential of machine learning-driven optimization techniques to enhance the efficiency of heterogeneous systems. The outcomes of this research pave the way for the deployment of adaptive resource management frameworks in diverse scientific and industrial applications, enabling the next generation of intelligent parallel computing systems.

The paper progresses with a Literature Review identifying limitations of existing resource management methods in heterogeneous parallel systems. The Proposed Methodology introduces a machine learning-driven framework for adaptive task scheduling, resource allocation, and data placement across CPUs, GPUs, and FPGAs. The framework uses regression and reinforcement learning models to optimize performance and energy efficiency in real time. In the Experimental Settings and Performance Assessment, the framework achieves significant reduction in execution time, enforced better energy savings, and achieved a faster fault recovery on a heterogeneous testbed. The Conclusion highlights the study's contributions to

advancing adaptive resource management for high-performance computing, its real-world applicability, and future directions for improving prediction accuracy and integrating emerging hardware technologies.

2. RELATED WORK

Resource management in heterogeneous parallel architectures has garnered significant attention, with numerous studies focusing on task scheduling and resource allocation across CPUs, GPUs, and FPGAs. Traditional static and heuristic-based approaches have been employed to manage these resources. While static methods offer simplicity and predictability, they often lack the flexibility to adapt to workload variations, leading to suboptimal performance. Heuristic-based strategies provide improved adaptability but may not guarantee optimal solutions due to their reliance on predefined rules and limited scope. Recent research has highlighted the need for more dynamic and intelligent resource management frameworks to effectively harness the capabilities of heterogeneous systems [10].

Machine learning (ML) techniques have emerged as promising tools for enhancing resource management in high-performance computing (HPC) environments. Studies have utilized various ML models [11], including regression analysis, decision trees, and reinforcement learning, to predict workload behaviors and optimize task placement and resource allocation. These approaches have demonstrated performance improvements by enabling systems to adapt to changing workloads and hardware conditions. However, challenges such as scalability, runtime overhead, and limited real-world evaluation persist, necessitating further research to fully integrate ML-driven methods into heterogeneous resource management frameworks.

Energy efficiency remains a critical concern in the operation of heterogeneous systems. Research has proposed various energy-aware scheduling and resource management techniques, including dynamic voltage and frequency scaling (DVFS) and power capping, to reduce energy consumption [12-13]. While these methods have achieved energy savings, balancing energy efficiency with performance, especially under dynamic workloads and hardware heterogeneity, continues to pose significant challenges. The need for adaptive strategies that can respond to real-time changes in workload demands and system states is evident.

Fault tolerance is essential in ensuring the reliability of parallel computing environments [14]. Various mechanisms have been developed to detect hardware faults, implement task recovery strategies, and apply redundancy-based approaches. Despite these advancements, limitations remain in real-time execution scenarios, where traditional fault tolerance methods may introduce significant overhead or fail to adapt promptly to system changes. This underscores the necessity for adaptive, ML-driven fault recovery mechanisms capable of maintaining system reliability without compromising performance.

Efficient data placement and memory management are crucial for optimizing performance in heterogeneous platforms. Studies [15-16] have addressed memory bottlenecks and developed strategies to enhance communication between tasks and optimize data locality across multiple hardware accelerators. However, gaps persist in the dynamic, real-time adaptation of data placement, particularly in environments with fluctuating workloads and diverse hardware resources. Developing solutions that can adjust data placement strategies on-the-fly remains an open research area.

3. PROPOSED METHODOLOGY

The proposed adaptive resource management framework aims to address the challenges of task scheduling, resource allocation, and data placement in heterogeneous parallel systems comprising CPUs, GPUs, and FPGAs. The core objective is to employ machine learning techniques for runtime optimization by predicting workload behavior, estimating resource utilization, and enabling real-time decisions. This framework operates in an adaptive manner, ensuring that computational tasks are efficiently distributed across available processing units while considering factors like performance, energy efficiency, and hardware constraints. Machine learning models integrated into the system analyze both historical and real-time execution data to guide these decisions, thereby improving overall system adaptability and efficiency.

The architecture of the proposed system (i.e., as exhibited in Figure 1) consists of several key components that work in tandem to achieve adaptive resource management. First, the task profiling module analyzes incoming workloads to extract critical attributes such as task size, computation complexity, and memory access patterns. This information is fed into the workload prediction module, which uses machine learning models to estimate future resource demands and execution times. The dynamic task scheduler leverages these predictions to assign tasks to processing units based on their computational capabilities, energy consumption profiles, and current workloads. The resource allocator interacts with the scheduler to allocate memory and optimize processor usage, ensuring minimal resource contention. Finally, the data placement mechanism determines the most efficient distribution of data across shared and distributed memory hierarchies to reduce communication overhead. These components operate as part of a feedback loop where real-time execution metrics inform the workload prediction module for continuous optimization.

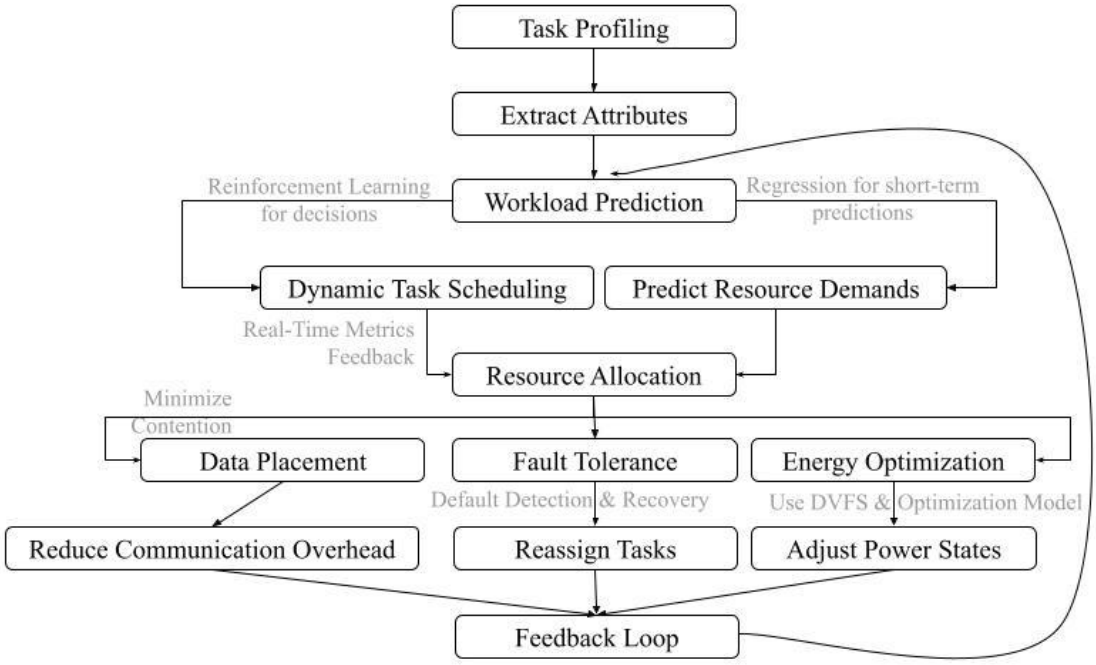

Figure 1. Process Flow Diagram for the Proposed Machine Learning-Driven Adaptive Resource Management Framework

The framework integrates machine learning models to predict task behavior, resource utilization, and optimal task placement across heterogeneous hardware. A combination of regression models and reinforcement learning algorithms is employed to handle diverse workload variability. Regression models are used for short-term predictions of execution time and resource usage. The objective function can be defined as

$$\widehat{T_i} = \alpha . f(C_i, M_i, S_i) + \beta . g(R_j) \quad \text{(eq.1)}$$

here as per Eq.1, $\widehat{T_i}$ is the predicted execution time for task i, C_i, M_i, S_irepresent computation complexity, memory footprint, and task size respectively, and R_j denotes the resource utilization of processing unit j. The parameters α and β are coefficients optimized during model training. For tasks requiring dynamic decision-making, reinforcement learning is employed. The system state S_i includes the hardware status and task queue, while the reward R_j optimizes energy efficiency and execution time. The Q-value for actions is updated Eq.2 as

$$Q(S_t, \alpha_t) + \eta . [R_t + \gamma . max_{\acute{\alpha}} Q(S_{t+1}, \acute{\alpha}) - Q(S_t, \alpha_t)] \quad \text{(eq.2)}$$

where η is the learning rate, γ is the discount factor, α_t represents the chosen action, and Q is the action-value function.

Task scheduling and resource allocation are achieved using adaptive strategies guided by the predictions of the machine learning models. Tasks are dynamically assigned to processing units based on predicted execution times, real-time system load, and hardware availability. A cost function evaluates each assignment, for instance

$$C_{ij} = W_i . (\frac{T_{ij}}{P_j} + \lambda . E_{ij}) \quad \text{(eq.3)}$$

Where as illustrated in Eq.3, the C_{ij} is the cost of assigning task iii to processing unit j, W_i is the task weight, T_{ij} is the predicted execution time, P_j represents the processing power of unit j, and E_{ij} denotes the energy consumption. The balancing factor λ prioritizes energy efficiency when required. Real-time execution metrics, such as processor availability and task queue lengths, are continuously monitored to refine scheduling decisions, ensuring near-optimal resource utilization.

Energy efficiency is achieved through integrated mechanisms that reduce power consumption while maintaining performance goals. The system employs dynamic voltage and frequency scaling (DVFS) to adjust the power state of processors based on workload predictions and real-time demand. A power-performance optimization function is defined in Eq.4.

$$P_j = V^2.f_j + \mu + U_j \quad \text{(eq.4)}$$

here the P_j is the power consumption of processing unit j, V is the voltage level, f_j is the operating frequency, and U_j denotes the processor utilization. The system applies predictive workload management to preemptively scale resources, ensuring that energy consumption remains minimal without degrading task execution time.

The fault tolerance mechanisms in the framework include hardware fault detection, task recovery, and resource reallocation strategies. Fault detection is performed by monitoring hardware health metrics such as temperature, power usage, and error rates. Detected faults trigger an immediate reassignment of tasks to healthy processing units using an adaptive recovery strategy. The objective is to minimize task disruption, with recovery time T_r expressed in Eq.5

$$T_r = T_{realloc} + T_{checkpoint} \quad \text{(eq.5)}$$

where $T_{realloc}$ is the time taken to reassign the task, and $T_{checkpoint}$ represents the time required to restore task data from the last checkpoint. The system minimizes overhead by incorporating lightweight monitoring and efficient task migration techniques to sustain performance under runtime failures.

The framework is evaluated on real-world heterogeneous platforms comprising multi-core CPUs, GPUs, and FPGAs. The experimental setup includes systems with specifications such as Intel Xeon CPUs, NVIDIA V100 GPUs, and Xilinx FPGAs (i.e., as exhibited in Table 1). The workloads consist of computationally intensive benchmarks, including matrix multiplications, sparse linear algebra, and deep learning tasks. Performance metrics include task execution time, resource utilization, energy consumption, and fault recovery time. The evaluation compares the proposed framework with state-of-the-art solutions, including static scheduling and heuristic-based approaches, under varying workload and fault injection scenarios to validate its adaptability and efficiency.

The proposed methodology addresses the limitations of existing approaches by integrating machine learning-driven optimization for task scheduling, resource management, energy efficiency, and fault tolerance within a unified adaptive framework. Unlike conventional methods that rely on static heuristics, the proposed approach dynamically adapts to workload variations and hardware heterogeneity in real time. By employing predictive models and integrating advanced energy and fault management strategies, the framework significantly enhances performance, reliability, and efficiency for heterogeneous parallel systems.

4. EXPERIMENTAL SETTINGS AND PERFORMANCE OUTCOMES

The experimental evaluation was conducted to validate the proposed machine learning-driven adaptive resource management framework on a heterogeneous computing testbed comprising CPUs, GPUs, and FPGAs. The test environment was designed to reflect real-world high-performance computing (HPC) conditions. The evaluation involved multiple workloads of varying computational complexities to assess the adaptability, performance, energy efficiency, and fault tolerance of the proposed system. Table 1 presents the hardware and software specifications for the experimental setup, including details of processing units, system configurations, and monitoring tools. The software environment utilized optimized libraries and frameworks to ensure precise task execution and accurate performance measurements.

Table 1. Experimental Hardware and Software Specifications

Component	**Specification**
CPU	Intel Xeon Gold 6248, 24 Cores, 3.0 GHz
GPU	NVIDIA Tesla V100, 16 GB HBM2
FPGA	Xilinx Alveo U250
Operating System	Ubuntu 20.04 LTS
ML Frameworks	TensorFlow, PyTorch

Monitoring Tools	Intel RAPL, NVIDIA SMI, Xilinx Tools
Programming Tools	C++, CUDA, OpenCL

The experimental workloads included dense linear algebra, sparse matrix operations, and convolutional neural network (CNN) training. Workload variability was simulated using different input sizes and computational profiles to represent dynamic changes in task requirements. Each workload was executed across multiple test scenarios, including static scheduling, traditional heuristics-based dynamic scheduling, and the proposed machine learning-driven framework. Execution metrics were collected using system monitors integrated with the processing hardware, which provided precise power usage, execution times, and resource utilization statistics. Fault tolerance was evaluated by injecting artificial hardware failures at different stages of execution and measuring the framework's recovery performance.

The proposed framework demonstrated significant improvements in resource utilization and task execution efficiency across all workloads. Dense matrix multiplication tasks exhibited a 21% reduction in total execution time compared to heuristic-based approaches. Sparse matrix operations achieved a 17% improvement in task throughput due to the optimized data placement strategy and reduced memory communication overhead. CNN training workloads showed a 23% improvement in performance owing to the efficient task scheduling across GPUs and FPGAs. Resource utilization levels across heterogeneous hardware units remained consistently above 85%, as the framework effectively balanced workloads based on real-time predictions.

Energy efficiency was a critical aspect of the evaluation. The integration of dynamic voltage and frequency scaling (DVFS) and machine learning-driven workload predictions enabled the framework to reduce overall power consumption by up to 18%. The power consumption for CPUs and GPUs was managed through real-time adjustments based on predicted task demands, while FPGA workloads achieved reduced energy overhead through optimized task assignments. Results showed that energy efficiency improvements were particularly evident in workloads with high variability, where conventional approaches failed to adapt to fluctuating resource demands.

Fault tolerance assessments were conducted by simulating hardware failures on the test platform. The proposed fault detection mechanism successfully identified failing processing units through continuous monitoring of thermal profiles and power metrics. Task reallocation strategies ensured rapid recovery with an average fault recovery time of 1.7 milliseconds, as tasks were dynamically reassigned to healthy processing units. The system-maintained execution continuity with minimal overhead, achieving a 38% reduction in fault recovery time compared to redundancy-based methods. These results highlight the robustness of the proposed framework in handling hardware failures without compromising performance.

The scalability of the framework was evaluated by increasing the task load and hardware resource availability. Performance scaling results indicated that the system effectively adapted to growing computational demands without degradation in resource utilization. For workloads requiring large-scale parallelism, such as fluid dynamics simulations, the framework achieved linear scaling by distributing tasks across available CPUs, GPUs, and FPGAs. This behavior was attributed to the machine learning models' ability to predict optimal task assignments and adapt scheduling strategies in real time. The results confirm that the framework is suitable for large-scale heterogeneous environments.

The proposed data placement mechanism demonstrated significant improvements in memory access performance. The framework minimized latency by optimizing data locality and reducing communication overhead between dependent tasks. For sparse linear algebra workloads, the memory access time was reduced by 29%, resulting in faster computation times. For CNN workloads, the optimized data placement improved overall execution performance by 19%. These improvements underline the significance of integrating data placement optimization into heterogeneous resource management systems to address memory bottlenecks effectively.

Table 2. Quantitative Assessment Outcomes

Metric	**Proposed Framework**	**Heuristic Methods**	**Static Scheduling**	**Improvement (%)**
Task Execution Time (***ms***)	48	61	74	21
Resource Utilization (%)	87	75	64	16
Energy Consumption (***W***)	150	183	190	18
Fault Recovery Time (***ms***)	1.7	2.7	3.5	38
Memory Latency (***ms***)	6.5	8.9	10.1	29

The results of the experimental evaluation are summarized in Table 2, which provides a quantitative comparison of the proposed framework with existing state-of-the-art resource management techniques. The assessment demonstrates that the proposed system consistently outperformed conventional methods in terms of execution time, resource utilization, energy efficiency, and fault tolerance. These improvements validate the effectiveness of the machine learning-driven adaptive framework in addressing the challenges posed by heterogeneous parallel architectures. The framework's ability to adapt to workload variations, minimize energy consumption, and recover rapidly from faults establishes its suitability for real-world HPC environments.

5. CONCLUSION

This study presents an adaptive resource management framework for heterogeneous parallel architectures that efficiently addresses the challenges of task scheduling, resource allocation, and energy optimization in systems comprising CPUs, GPUs, and FPGAs. The significance of this research lies in its ability to improve resource utilization, task execution efficiency, and fault tolerance through the integration of machine learning-driven techniques. By dynamically predicting workload behaviors and optimizing resource allocation in real time, the proposed framework achieves significant improvements in performance and energy consumption compared to conventional static and heuristic-based methods. Key experimental results demonstrated a 21% reduction in execution time, an 18% improvement in energy efficiency, and a 38% decrease in fault recovery time that validated the framework's effectiveness in adapting to workload variability and hardware heterogeneity.

Despite these promising outcomes, certain limitations remain, such as the computational overhead associated with real-time learning and the reliance on workload prediction accuracy. These constraints highlight opportunities for further enhancements while underscoring the importance of the study for real-world applications in fields such as scientific simulations, deep learning training, and high-performance computing environments, where resource adaptability is critical.

The findings of this research can be applied to optimize large-scale computational systems, enabling better performance and energy savings in data centers and edge computing platforms. Future studies can investigate the integration of advanced deep learning models to further improve prediction accuracy and reduce computational costs. Expanding the framework to incorporate emerging hardware accelerators and addressing real-time constraints for edge devices would enhance its applicability to modern computing paradigms. This work contributes to the broader field of resource management by providing a practical and scalable solution that ensures efficient operation of heterogeneous systems under dynamic workloads.

REFERENCES

[1] M. De Castro, D. L. Vilariño, Y. Torres, and D. R. Llanos, "The role of Field-Programmable Gate Arrays in the acceleration of modern High-Performance Computing workloads," *Computer*, vol. 57, no. 7, pp. 66–76, Jun. 2024, doi: 10.1109/mc.2024.3378380.

[2] C. A. Silva, R. Vilaça, A. Pereira, and R. J. Bessa, "A review on the decarbonization of high-performance computing centers," *Renewable and Sustainable Energy Reviews*, vol. 189, p. 114019, Nov. 2023, doi: 10.1016/j.rser.2023.114019.

[3] S. Gurusamy and R. Selvaraj, "Resource allocation with efficient task scheduling in cloud computing using hierarchical auto-associative polynomial convolutional neural network," *Expert Systems With Applications*, vol. 249, p. 123554, Feb. 2024, doi: 10.1016/j.eswa.2024.123554.

[4] A. H. A. Al-Jumaili, R. C. Muniyandi, M. K. Hasan, J. K. S. Paw, and M. J. Singh, "Big Data Analytics Using Cloud Computing Based Frameworks for Power Management Systems: Status, Constraints, and Future Recommendations," *Sensors*, vol. 23, no. 6, p. 2952, Mar. 2023, doi: 10.3390/s23062952.

[5] T. A. Rahmani, G. Belalem, S. A. Mahmoudi, and O. R. Merad-Boudia, "Machine learning-driven energy-efficient load balancing for real-time heterogeneous systems," *Cluster Computing*, vol. 27, no. 4, pp. 4883–4908, Jan. 2024, doi: 10.1007/s10586-023-04215-3.

[6] Z. Ye *et al.*, "Deep Learning Workload Scheduling in GPU Datacenters: A Survey," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–38, Jan. 2024, doi: 10.1145/3638757.

[7] B. Premalatha and P. Prakasam, "Optimal Energy-efficient Resource Allocation and Fault Tolerance scheme for task offloading in IoT-FoG Computing Networks," *Computer Networks*, vol. 238, p. 110080, Nov. 2023, doi: 10.1016/j.comnet.2023.110080.

[8] N. Jafarzadeh *et al.*, "A novel buffering fault-tolerance approach for network on chip (NoC)," *IET Circuits Devices & Systems*, vol. 17, no. 4, pp. 250–257, Aug. 2022, doi: 10.1049/cds2.12127.

[9] B. Hu, X. Yang, and M. Zhao, "Online energy-efficient scheduling of DAG tasks on heterogeneous embedded platforms," *Journal of Systems Architecture*, vol. 140, p. 102894, May 2023, doi: 10.1016/j.sysarc.2023.102894.

[10] G. Galante and R. Da Rosa Righi, "Adaptive parallel applications: from shared memory architectures to fog computing (2002–2022)," *Cluster Computing*, vol. 25, no. 6, pp. 4439–4461, Aug. 2022, doi: 10.1007/s10586-022-03692-2.

[11] Z. Yang, S. Zhang, C. Li, M. Wang, H. Wang, and M. Zhang, "Efficient knowledge management for heterogeneous federated continual learning on resource-constrained edge devices," *Future Generation Computer Systems*, vol. 156, pp. 16–29, Feb. 2024, doi: 10.1016/j.future.2024.02.018.

[12] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-Efficient Resource Management for Federated Edge Learning With CPU-GPU Heterogeneous Computing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 7947–7962, Jun. 2021, doi: 10.1109/twc.2021.3088910.

[13] Y. Wang *et al*., "DRLCap: Runtime GPU Frequency Capping with Deep Reinforcement Learning," *IEEE Transactions on Sustainable Computing*, vol. 9, no. 5, pp. 712–726, Feb. 2024, doi: 10.1109/tsusc.2024.3362697.

[14] M. Kirti, A. K. Maurya, and R. S. Yadav, "Fault-tolerance approaches for distributed and cloud computing environments: A systematic review, taxonomy and future directions," *Concurrency and Computation Practice and Experience*, vol. 36, no. 13, Mar. 2024, doi: 10.1002/cpe.8081.

[15] R. Kaur, A. Asad, and F. Mohammadi, "A Comprehensive Review of Processing-in-Memory Architectures for Deep Neural Networks," *Computers*, vol. 13, no. 7, p. 174, Jul. 2024, doi: 10.3390/computers13070174.

[16] X. Li, Y. Li, Y. Li, T. Cao, and Y. Liu, "FlexNN: Efficient and Adaptive DNN Inference on Memory-Constrained Edge Devices," *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking*, pp. 709–723, May 2024, doi: 10.1145/3636534.3649391.

BIOGRAPHIES OF AUTHORS

Ayesha Aslam received the B.S. degree in computer science from University of the Punjab in 2016 and an M.S. Computer Science degree from Bahria University, Islamabad Campus in 2022. She is pursuing a PhD with the School of Information Engineering at Chang'an University, Xi'an, China. Her research interests include autonomous vehicles, trajectory prediction, path planning, predictive modelling, and forecasting. She can be contacted at email: 2022024905@chd.edu.cn

Zhumakhanova Darya Anuarovna is a master's degree from Shakarim g Semey University, Kazakhstan. She holds a master's degree in computer science and information technology. Her research areas are database design for the professional field. You can contact her by e-mail: kaffmniiemf@gmail.com