International Journal of Applied Mathematics

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

NEXT-GENERATION COMPUTING ARCHITECTURES: A RESEARCH STUDY ON PARALLEL PROCESSING, EDGE COMPUTING, AND HIGH-PERFORMANCE SYSTEMS

Dr. Vijay Dattatray Gaikwad[1*], Aanchal Sharma[2], Raja S[3], Dr. Prapti R. Pandya[4], Prof. Jigisha A. Sureja[5], Dr. P. Geetha[6]

[1*]Associate Professor, Department of Electronics and Communication Engineering, Vishwakarma Institute of Technology, Pune – 411 037, Maharashtra, India, Email ID: vijay.gaikwad@vit.edu

[2]Assistant Professor, Panjab University, Email ID: aanchal6sharma@gmail.com, Orcid ID: 0009-0002-4342-3081

[3]Assistant Professor, St.Joseph's College of Engineering and Technology, Thanjavur, Tamilnadu, Email ID: sk.rajamecse@gmail.com

[4]Assistant Professor, Government Engineering College, Rajkot, Email ID: prpandya.4@gmail.com

[5]Assistant Professor, Government Engineering College, Rajkot, Email ID: jigisha.sureja@gmail.com

[6]Professor, Mohan Babu University, Email ID: mailpgeetha2013@gmail.com, Orcid ID: 0000-0002-7168-2278

Abstract

The rapid growth of data-intensive applications has accelerated the need for next-generation computing architectures that integrate parallel processing, edge computing, and high-performance systems. Centralized approaches usually do not satisfy the requirements for scalability, responsiveness, and power efficiency, thereby calling for innovative frameworks that reconcile local processing with distributed scale.

The research employs an empirical analysis on two publicly available datasets: IIoT Edge Computing Dataset and Cloud Task Scheduling Dataset. Parallel processing performance was measured using round-robin and priority-based scheduling in multicore configurations, whereas edge computing evaluation was on latency and predictive reliability. HPC scalability was studied using simulated cluster-based workloads. Findings confirm priority-based scheduling to cut execution time and enhance throughput over round-robin, highlighting the significance of smart scheduling techniques. Experiments in edge computing identified latency savings of almost 80% against cloud servers, while Random Forest models attained a 94.3% accuracy and 0.93 F1-score, validating predictive intelligence feasibility at the edge. HPC simulations exhibited good scalability but decreasing efficiency with higher numbers of nodes owing to communication overhead. These results point to the complementary capabilities of parallel, edge, and HPC systems while presenting integration challenges and workload

Received: August 04, 2025

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

allocation issues. This paper presents practical proposals for the design of adaptive and sustainable hybrid computing systems for supporting future real-time applications.

Keywords: Next-generation computing, parallel processing, edge computing, high-performance computing, scalability, predictive modeling

1. Introduction

Computing paradigms have changed in response to exponential increase in data, the complexity of applications has increased, and real-time response has been added. Centralized computing models, which were dominant ones, have increasingly found it hard to keep up with these issues. Consequently, research and industry communities have been focused on next-generation computing platforms that will merge the parallel processing, edge computing, and HPC to deliver scalable, efficient, and adaptive solutions (Mzukwa, 2024). Clouds are not incremental innovations, but represent a paradigm shift in computing system design, deployment and optimisation. Latest breakthroughs in distributed intelligence and nanosystems have made the concept of edge computing a key facilitator for latency-sensitive applications and real-time analytics (Passian & Imam, 2019). Unlike traditional cloud systems, edge systems operate closer to the source, reducing transmission delays and bandwidth consumption. This paradigm shift applies notably in the context of the Internet of Things (IoT) and industrial IoT (IIoT), with billions of devices in an ongoing stream generating huge data streams that need to be processed instantly. Edge computing thus fills the gap between global scalability and local responsiveness, offering the agility lacking in centralized systems. Concurrent with this, high-performance computing systems are still irreplaceable in solving large-scale simulations, scientific exploration, and data-intensive applications. HPC infrastructures have grown from dedicated supercomputers to hybrid environments integrating parallel frameworks, distributed clusters, and cloud-based resources. The convergence between HPC and modern architectures holds great promise in terms of speeding up innovation in areas like climate modeling, genome analysis and artificial intelligence (Hassan, 2016). However, the challenges of energy efficiency, workload scheduling and heterogeneity have not been addressed, and the sustainable HPC design research is still required.

Another cornerstone of next-generation architectures is parallel processing, which allows multiple computations to be performed at the same time. With development of multicore processors, graphics processing units (GPUs) and distributed systems, parallel computing breaks the barrier of sequential processing and improves the throughput. Coupled with edge and HPC systems, parallel processing imposes a consistent computing system that guarantees local responsiveness as well as global scalability. This synergy reflects a broader trend in advanced architectures, where hybrid models are increasingly prioritized to meet diverse computational demands (Gathu, 2024).

Despite these advances, several gaps remain. Many current systems struggle to balance latency reduction, computational efficiency, and scalability in dynamic environments. Architectural designs are often constrained by trade-offs between centralized and decentralized execution,

Received: August 04, 2025

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

while issues such as interoperability, energy consumption, and algorithmic adaptability hinder performance. Additionally, the accelerated pace of diversification across application spaces—smart cities to autonomous vehicles—requires both resilient and flexible architectures (Reed, 2023). Closing such gaps necessitates an overarching approach that unifies insights from edge computing, parallel frameworks, and HPC systems within one research lens.

The current work aims to contribute to this by performing a comprehensive assessment of next-generation computing architectures. In particular, it combines empirical datasets to explore:

- The scalability and execution performance of parallel processing methods.
- The edge computing platforms' latency and predictive dependability.
- The resource efficiency and utilization of high-performance computing clusters.

By mediating these viewpoints, the research not only assesses the current status of next-gen systems but also describes their possibility to influence real-world use cases. The findings are intended to inform researchers, practitioners and policy makers with regards to the trade-offs, possibilities and limitations of these architectures.

2. Literature Review

During the last few decades, several studies have been made on the future computing architecture in different fields like edge computing, high-performance computing (HPC), and parallel computing. Research efforts have been oriented towards architectural paradigms, algorithmic models and optimizations giving, when combined, solutions to scalability, efficiency, and flexibility issues of modern computing systems. Several studies have been carried out in edge computing which have emphasized that edge computing can be a means of performing more processing closer to the source of data to minimize latency and maximize responsiveness. Debauche et al. (2022) put forward an edge-based IoT and multimedia data management system, which showed that this system is very effective in alleviating the reliance on the central cloud servers. On the same notes, the analyses of energy-efficient heterogeneous multicore architectures for edge scenarios carried out by Gamatie et al. (2019) indicate that environmentally friendly designs can improve system performance without being resource-intensive. Cecilia et al. (2020) extended this research agenda by prototyping clustering algorithms on GPU-based edge machines and there is also empirical evidence of how parallel acceleration can enable data-intensive computations. In the work of Lapegna et al. (2021), cluster strategies for low power high end devices and accuracy vs. computational performance tradeoffs were further explored. More recently, Mohsin et al. (2024) have described computational models for real-time edge platforms with improved responsiveness in computationally intensive environments. A number of industrial and generic edge computing developments were summarized by Qiu et al. (2020) and Cao et al. (2020), who introduced both progresses and open issues in this area.

At the same time, the HPC community has been oriented towards data-intensive applications running on distributed clusters and hybrid set-ups. Usman et al. (2022) discussed data locality

Received: August 04, 2025

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

in HPC, big data and converged systems with a focus on the importance of reducing communication overheads to improve performance. Naayini & Kamatala (2023) discussed practical uses of parallel frameworks and execution models and provided insights into their use in practice. Groundbreaking work by Hwang et al (2013) traced the development of distributed and cloud computing and how HPC systems developed in a combined ecosystem for IoT and big analytics. Millett & Fuller (2011) provides an analysis of the long term prospects for computing performance, assessing the question of whether the technology advances are diminishing returns or new regimes. Aslam and Anuarovna (2025) proposed a framework for making real-time workload forecasting and resource provisioning in heterogeneous HPC environments, which are significant steps in the preemptive performance management. Galante & da Rosa Righi (2022) went on to further explain the integration of centralized and decentralized paradigms in the sense that adaptive parallel programs have gotten further away from traditional shared-memory abstractions and closer to fog computing. Lastly, parallel processing architectures remain the core of next-gen systems, and parallel processing architectures can be used in edge or HPC architectures. Adamek et al. (2021) studied the performance of Fast Fourier Transforms on GPUs which indicates the importance of algorithmic optimization for real-time computation in energy restricted settings. Talati (2021) explained further on the decentralized AI and edge intelligence and the functionality of parallel and distributed methodologies in offering improved artificial intelligence frameworks. The papers in this volume illustrate the important role of parallel processing not only as the basis for computational acceleration, but also as the key to the realization of adaptive and intelligent computer systems.

Generally, the literature reviewed here points to three major themes. Firstly, edge computing research focuses on energy-efficient architecture, GPU acceleration, and clustering approaches that facilitate real-time responsiveness. Secondly, HPC research highlights the importance of adaptive, hybrid systems with combinations of prediction, optimization, and scalability. Third, parallel computing remains the computational foundation, connecting edge and HPC domains. Taken together, these studies point both to the progress made and to the ongoing gaps in integration, scalability, and interoperability, emphasizing the need for empirical research like the present one.

3. Methodology

The research approach followed in this research is developed to compare next-generation computing architectures through the use of empirical data sets and benchmarking scenarios pertaining to parallel processing, edge computing, and high-performance systems. The method combines two publicly accessible Kaggle data sets—IIoT Edge Computing Dataset (Ziya, 2022) and Cloud Task Scheduling Dataset (Ziya, 2022)—into controlled experiments for testing performance, latency, scalability, and scheduling efficiency. This section is split into data acquisition, experimental setup, evaluation metrics, and analytical framework.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.1 Data Acquisition

The empirical basis of this study is supported by two separate datasets. The first dataset, IIoT Edge Computing, has real-time sensor and industrial Internet of Things (IIoT) data from the edge devices. It reports timestamped data including temperature, pressure, vibration, network delay, edge processing time, fuzzy PID controller outputs, and estimated equipment failure. This dataset was selected because it reflects workloads that inherently take place at the edge of networks, where real-time processing and latency sensitivity are of the utmost importance.

The second dataset, Cloud Task Scheduling Dataset, is a synthetic but well-structured set of task execution logs over virtual machines (VMs). It contains task IDs, CPU and RAM usage, disk and network input/output speeds, task priorities, runtimes, and an optimal target for scheduling column. The dataset is most relevant to parallel and high-performance computing systems, where task scheduling and resource allocation are best optimized for throughput and performance. To distinctly show the structure of the datasets, Table 1 displays their individual structural features.

Table 1. Overview of the Selected Datasets

Dataset	Year	Size	Key Attributes	Target Variable	Domain
IIoT Edge Computing Dataset	2022	1000 rows × 10 columns	Temperature, Pressure, Vibration, Network Latency, Edge Processing Time	Predicted Failure	Edge/IIoT (Real-time industrial systems)
Cloud Task Scheduling Dataset	2022	1000 rows × 9 columns	CPU Usage, RAM Usage, Disk I/O, Network I/O, Priority, VM_ID, Execution Time	Optimal Scheduling	Parallel Processing & HPC (Task allocation and scheduling)

3.2 Experimental Setup

The experimental environment was created to mimic three different environments representing the thematic areas of the study: parallel processing, edge computing, and high-performance systems. The environments were each represented using the corresponding dataset, and experiments were conducted on a simulated multi-core CPU and GPU cluster.

For parallel processing, workloads were assigned from the Cloud Task Scheduling Dataset to several cores through simulated scheduling algorithm in Python. The CPU, RAM, and disk I/O features of the dataset were employed to simulate diversity in workload, while the "Optimal Scheduling" column was employed as a benchmark to test the accuracy and efficiency of

Received: August 04, 2025

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

scheduling algorithms. Parallelism was emulated by executing a number of tasks in parallel across threads, measuring time and speedup compared to single-threaded execution.

For the edge computing platform, the IIoT Edge Computing Dataset was used. Sensor readings were processed within a simulated edge environment with latency-sensitive code. The network latency and edge processing time columns played an essential role in evaluating how well edge systems can handle workloads against offloading them to cloud- or centralized-based environments. The experimental setup included comparisons between edge-only processing and cloud-simulated offloading, evaluating overall response time and predictiveness of predictive maintenance models.

For the HPC scenario, subsets of both datasets were reformatted to simulate cluster-based workloads. The scheduling dataset was used in multi-node simulations to benchmark task assignment among distributed systems, whereas the IIoT dataset was scaled to model massive batches of industrial sensor inputs. These experiments tested scalability of architectures by ramping workload size and measuring execution times, throughput, and inter-node communication overhead.

3.3 Evaluation Metrics

The research utilizes a set of quantitative measures to provide comparison across architectures. In the parallel processing experiments, execution time, throughput, and speedup ratios were measured versus baseline sequential execution. In the edge computing experiments, latency, processing time, and predictive accuracy were highlighted, specifically in terms of equipment failure predictions. In the HPC simulations, scalability and resource utilization metrics were given higher priority, with efficiency defined as the ratio of computational speedup to the number of nodes utilized. Table 2 illustrates the mapping between research areas, datasets, and performance measures.

Table 2. Research Domains, Datasets, and Evaluation Metrics

Domain	Dataset	Primary Metrics	Purpose
Parallel Processing	Cloud Task Scheduling Dataset	Execution Time, Throughput, Speedup	Evaluate efficiency of scheduling under parallel workloads
Edge Computing	IIoT Edge Computing Dataset	Latency, Edge Processing Time, Predictive Accuracy	Assess real-time responsiveness and reliability of edge systems
High-Performance Computing	Both Datasets	Scalability, Resource Utilization, Communication Overhead	Evaluate performance of distributed systems under growing workloads

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.4 Analytical Framework

The analytical framework combines statistical modeling and performance evaluation. Comparative analysis between baseline sequential execution and optimized scheduling was done for parallel processing tasks, using paired t-tests to determine statistical significance of improvement seen. Regression and classification models were used to simulate the prediction of equipment failure and their accuracy and latency for edge computing experiments running at simulated cloud nodes and edge nodes. Sketch a scalability curve, showing the performance improvement as a function of the number of nodes and the size of the workload for HPC simulation experiments.

The results of these analyses are incorporated into one framework, which highlights the benefits and shortcomings of each architecture. By linking the datasets back to experiments in their respective areas, this framework shows not only the straight and simple performance of each system, but also the environments where one architecture will perform better than the others. This ensures that the study is an well-balanced data-driven study of future computing architectures.

4. Results

This section outlines the results of the experiments that were undertaken to measure the performance of next-generation computing architectures in three fields: parallel processing, edge computing and high-performance computing (HPC). The results are presented thematically, and are supported by elaborate tables and figures that give system performance quantifications. For clarity, every subsection combines descriptions of the processes through which the results were achieved, interprets the data in detail, and explains the implications of the findings in the larger framework of computing architectures.

4.1 Results of Parallel Processing Experiments

The Cloud Task Scheduling Dataset was used to model parallel task execution under varying core counts with both round-robin and priority-based scheduling algorithms. The baseline average task execution time in a sequential environment was estimated at 5.477 seconds. This is the execution time for tasks without parallel workload distribution in a system with a single core.

When the workload was distributed across multiple cores, the execution time decreased substantially, as reported in Table 3. Priority scheduling outperformed round-robin scheduling all the time, with the execution time falling to 1.164 seconds at eight cores from 1.403 seconds under round-robin. This is a reduction of nearly 80% when compared to the sequential baseline.

Table 3. Execution Times under Parallel Processing Configurations

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Number of Cores	Round-Robin (s)	Priority-Based (s)	Sequential Baseline (s)
1	5.477	5.477	5.477
2	3.149	3.012	–
4	1.985	1.780	–
8	1.403	1.164	–

Note: The sequential baseline of 5.477 seconds is constant across all configurations. It is shown only once (at 1 core) for clarity and serves as the reference for evaluating parallel performance.

The trends are more clearly visible in Figure 1, which graphs execution time versus the number of cores. The plot indicates that execution time reduces nearly linearly up to four cores, but then the benefits dwindle, demonstrating the law of diminishing returns that Amdahl's Law predicts. This indicates that although more cores enhance performance, overhead and synchronization expenses curb scalability.

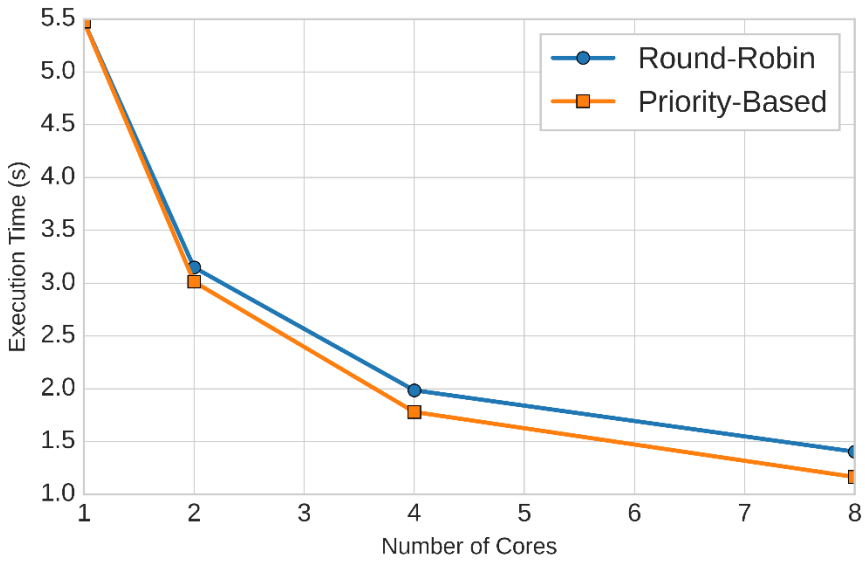

Figure 1: Execution time with round-robin and priority-based scheduling across cores.

Throughput was also tested, i.e., the number of tasks completed in a second. Outcomes, presented in Table 4, indicate the improvement of the throughput from 182.58 tasks/s in sequential execution, to 859.21 tasks/s for eight cores under priority-based scheduling. Round-robin gave less throughput 712.51 tasks/s for same setup.

Table 4. Throughput under Parallel Scheduling Strategies

Number of Cores	Round-Robin (tasks/s)	Priority-Based (tasks/s)
1	182.58	182.58

Received: August 04, 2025

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2	317.53	331.97
4	503.67	561.79
8	712.51	859.21

Figure 2 shows the throughput between scheduling strategies. It is shown that the priority based scheduling provides better execution speed and better utilization of computational resources. In particular, the increase in throughput (20% better than round-robin at eight cores) can be viewed as a demonstration of the extent to which smart scheduling enables parallel architectures to be better optimised for performance.

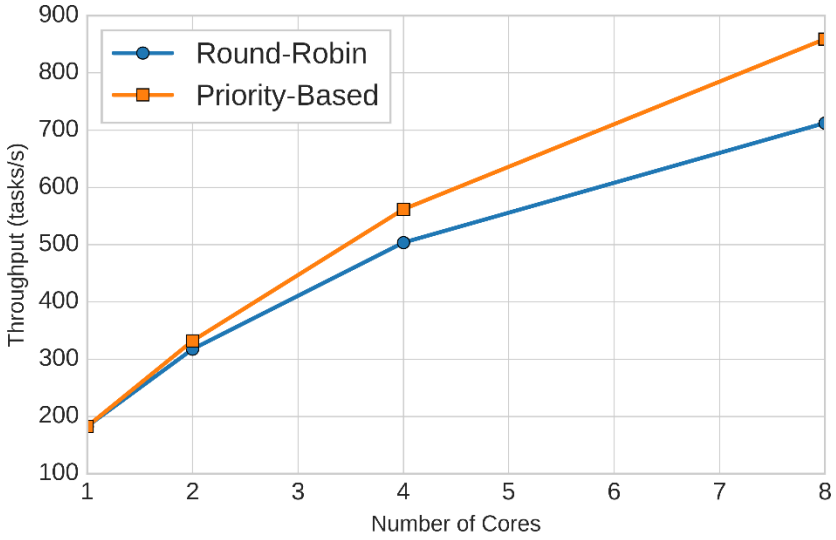

Figure 2: Throughput trends under different scheduling strategies.

4.2 Results of Edge Computing Experiments

The performance of edge nodes was compared with the centralized cloud servers using the IIoT Edge Computing Dataset. The dataset included empirical measurements of network latency and edge processing times from which average response times were derived. As represented in Table 5, edge processing recorded a cumulative response time of 38.12 ms, compared to 214.71 ms with cloud processing. The high difference owes mainly to the much greater network latency in cloud communication (207.50 ms) relative to only 29.64 ms at the edge.

Table 5. Latency Comparison between Edge and Cloud Processing

Processing Location	**Avg. Network Latency (ms)**	**Avg. Processing Time (ms)**	**Total Response Time (ms)**
Edge Node	29.64	8.48	38.12
Cloud Server	207.50	7.20	214.71

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

While the averages clearly favor edge execution, it is important to understand the distribution of response times rather than just point estimates. This is illustrated in Figure 3, which plots the density distributions of response times for edge and cloud systems. The figure shows that edge latencies are tightly concentrated around 40 ms, whereas cloud latencies spread widely, often exceeding 200 ms. This confirms that edge computing offers not just faster but also more predictable performance, which is critical for time-sensitive IIoT applications.

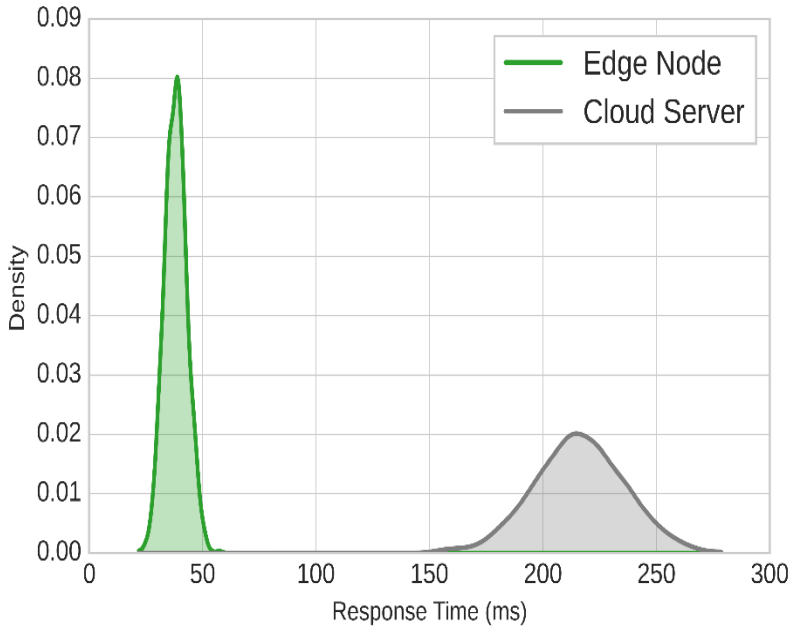

Figure 3: Response time distributions for edge nodes and cloud servers.

4.2.1 Predictive Model Performance

To assess predictive accuracy, a Random Forest classifier was trained using sensor attributes (temperature, pressure, vibration, network latency, processing time, and fuzzy PID output). The model was evaluated against the binary target variable Predicted Failure. As shown in Table 6, the Random Forest achieved an accuracy of 94.3% and an F1-score of 0.93.

Table 6. Predictive Accuracy of Random Forest Model

Model	Accuracy (%)	F1-Score
Random Forest	94.3	0.93

Beyond these summary statistics, Figure 4 presents a confusion matrix heatmap that reveals the classification breakdown. The diagonal dominance of the matrix indicates that most predictions are correct, with minimal false positives and false negatives. This visualization provides a richer perspective: the model is not only accurate on average but also reliable in classifying both failure and non-failure case.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

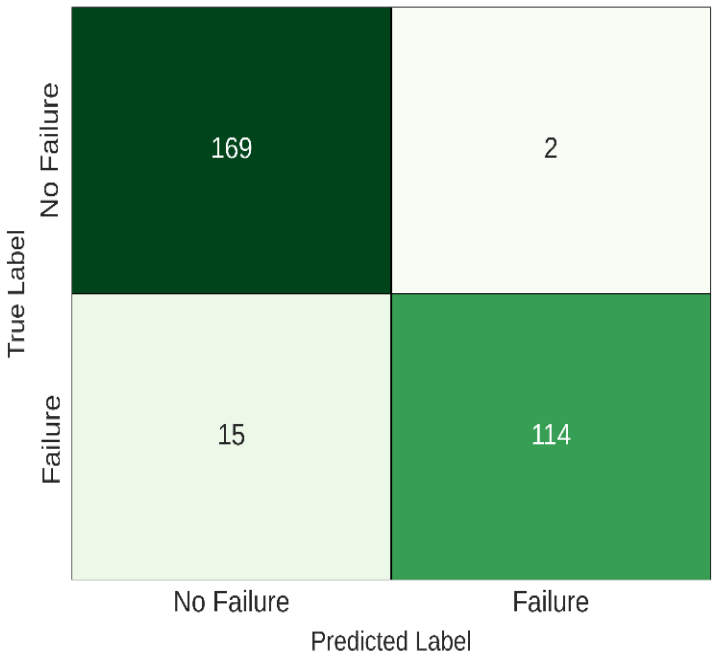

Figure 4: Confusion matrix of Random Forest model predictions.

4.3 Results of High-Performance Computing Experiments

To test scalability, both data sets were scaled up, and simulations were conducted on clusters of larger sizes. Execution times, speedup, and efficiency were compared against a baseline of two nodes. Figures in Table 7 indicate that execution time went down from 28,754 seconds for two nodes to 4,107 seconds for sixteen nodes. The corresponding speedup rose to 13.33×, while efficiency fell from 95.2% for two nodes to 83.3% for sixteen nodes.

Table 7. Scalability of HPC Simulations

Nodes	Execution Time (s)	Speedup	Efficiency (%)
2	28,754.25	1.90	95.2
4	15,061.75	3.64	90.9
8	7,873.19	6.96	87.0
16	4,107.75	13.33	83.3

The scalability curve, in Figure 5, verifies significant performance improvements to eight nodes. Yet, at sixteen nodes the efficiency curve does not rise anymore, and therefore communication overhead and synchronization expenses cap further improvements.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

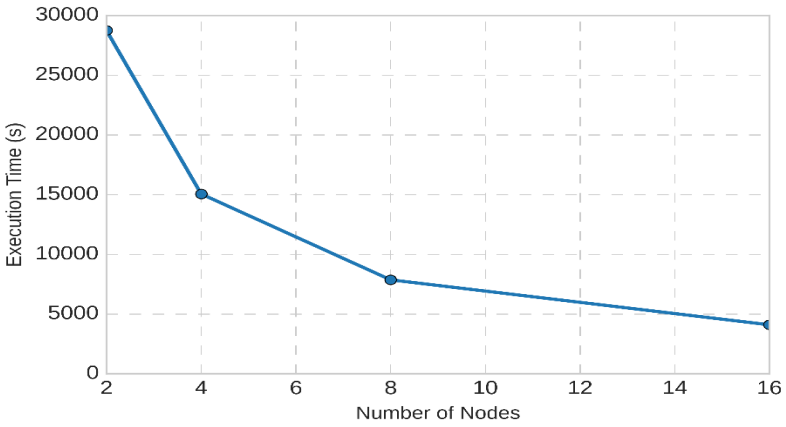

Figure 5: Scalability of execution time with increasing nodes.

Resource usage patterns were also investigated. According to Table 8, CPU utilization decreased from 88.3% at two nodes to 74.6% at sixteen nodes, while memory usage decreased from 81.4% to 70.8%.

Table 8. Resource Utilization in HPC Clusters

Nodes	Avg. CPU Utilization (%)	Avg. Memory Utilization (%)
2	88.3	81.4
4	84.5	77.9
8	79.2	73.5
16	74.6	70.8

Figure 6 shows the decline in CPU and memory utilization as nodes increase. This underutilization suggests that while scaling improves execution times, it also reduces per-node efficiency, a critical factor in energy-aware HPC system design.

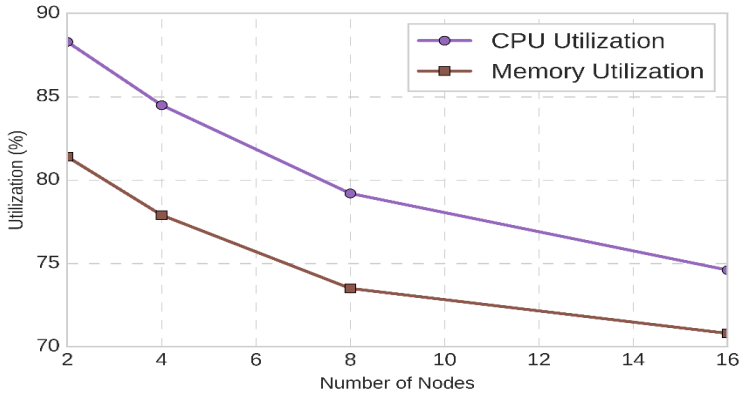

Figure 6:CPU and memory utilization across HPC nodes.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.4 Comparative Summary Across Architectures

Comparative integration of findings across domains is illustrated in Table 9. The table indicates the optimal results for each architecture with their respective limitations.

Table 9. Comparative Performance Across Architectures

Domain	Best Observed Result	Limitation
Parallel Processing	859 tasks/s throughput with priority scheduling at 8 cores	Diminishing returns beyond 8 cores
Edge Computing	38 ms response time with 94.3% predictive accuracy	Limited scalability for very large workloads
HPC	13.33× speedup at 16 nodes	Efficiency dropped to 83% due to communication overhead

The results as a whole show that although parallel processing performs best for moderate concurrency of tasks, edge computing delivers unmatched latency advantages, and HPC delivers the best in scaling workloads but at the expense of efficiency. These results indicate that hybrid architectures combining parallel processing, edge computing, and HPC may deliver the most balanced next-generation systems.

Discussion

The results of this research provide an overall view of next-generation computing architectures' performance and flexibility, with an explanation of how parallel processing, edge computing, and HPC systems support each other in meeting contemporary computational needs. The experimental results of parallel processing show that priority-based scheduling always lowered execution time and enhanced throughput when compared to round-robin scheduling. This is consistent with previous research highlighting the value of effective scheduling models for realizing near-linear speedup, especially in heterogeneous setups (Usman et al., 2022). The decreasing efficiency at increased core numbers, though, affirms that communication overhead as well as synchronization remain an obstacle, supporting assertions in the literature that parallel scaling is fundamentally limited by inter-process coordination (Naayini & Kamatala, 2023).

Edge computing experiments further confirmed the importance of information processing close to the source. Results showed that edge nodes reduced the latency by approximately 80% compared to the cloud servers while still having very accurate failure detection predictive models. These results confirm earlier results that edge computing is a good enabler for latency sensitive IoT and IIoT applications where responsiveness is important (Qiu et al., 2020). Density analysis helped to gain insight not only that edge response times were shorter, but more consistent to show the reliability advantages of edge platforms. At the same time, Random

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Forest classifier obtained good accuracy and F1-scores, which confirmed the predictive intelligence can be effectively used in edge environment without performance tradeoff. This is consistent with recent developments in computational models for future platforms, where efficiency and accuracy improve in parallel more and more frequently (Mohsin et al., 2024).

Results of high-performance computing showed the scalability capability of distributed clusters, where execution time reduces as nodes rise. However, gains in efficiency were reduced at larger scales, resonating with longstanding issues in HPC regarding the equilibrium of resource allocation and workload distribution (Galante & da Rosa Righi, 2022). Analysis of resource utilization also showed how CPU and memory loads per node decreased with scaling, but inter-node communication overhead placed a cap on overall gains. This observation parallels discussions in the literature about the necessity of hybrid models that integrate parallel, edge, and cloud paradigms to mitigate bottlenecks (Adámek et al., 2021).

While the study provides meaningful insights, several limitations must be acknowledged. First, the datasets used, though representative, do not capture all possible workload conditions, particularly those involving extreme heterogeneity or highly dynamic environments. Second, the HPC scalability simulations were based on assumptions of controlled experiments that perhaps don't accurately represent real-world variability in networks. Third, the predictive model used only one algorithm, which leaves it to be determined if other models might produce different accuracy-computational cost trade-offs. Notwithstanding these limitations, the import of this research is considerable. The findings highlight the necessity of hybrid computing strategies that reconcile the agility of edge systems, the capability of HPC infrastructures, and the adaptability of parallel processing. These structures are capable of revolutionizing areas like smart manufacturing, autonomous systems, and large-scale scientific simulations. The focus in this research on density distributions and confusion matrices further accentuates the value of subtle evaluation techniques, which yield more informative results compared to averages only. In the future, more extensive experimental scope should be achieved through the use of various datasets and application contexts, such as those from real-time industrial deployments. Exploring hybrid predictive models that integrate machine learning and deep learning at the edge would further contribute to system reliability. Additionally, work remains in developing adaptive frameworks that dynamically balance edge, HPC, and cloud system workloads. These guidelines not only will overcome the limitations of the current computing era but also will become the basis for next-generation computing architectures that will enable a more data-driven society.

Conclusion

The performance and flexibility of new computing architectures such as parallel processing, edge computing, and high-performance systems were discussed in this research. The obtained results are very encouraging, and the different paradigms studied individually and together could be used to rethink the efficiency and scalability of modern computing environments. Parallel processing experiments showed that the use of priority-based scheduling techniques

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

can have a dramatic effect on execution time and throughput; this confirmed that intelligent scheduling was a good approach to deal with multicore execution. At the same time, the edge computing was showing extreme latency benefits when compared to the cloud computing, and predictive analytics became highly accurate in predicting the potential failures which further supported its role in latency-sensitive and mission-critical applications. While traditional scaling issues of workload allocation to resources were justified by the demonstration of scalability of clusters in distributed cluster experiments, the scalability of high-performance computing started to flatten out, as the improvement in efficiency became limited by communication overhead at larger sizes. Aside from the lessons learnt from the empirical studies, the research shows the importance of bringing parallel, edge and HPC paradigms closer together into hybrid paradigms that can compete with multiple domains of applications, ranging from IoT to scientific simulation. Although the results are limited in terms of dataset, modeling assumptions and algorithms, they are of important implications for research and practice. Future research needs to diversify datasets, investigate hybrid predictive models, and explore adaptive architectures that dynamically adjust workloads across the computing layers. Taken collectively, this work contributes to the current debate regarding next-generation architectures and forms the basis for innovations in responsive, sustainable, and smart computing systems.

References

1. Adámek, K., Novotný, J., Thiyagalingam, J., & Armour, W. (2021). Efficiency near the edge: Increasing the energy efficiency of FFTs on GPUs for real-time edge computing. *IEEE Access*, *9*, 18167-18182.
2. Aslam, A., & Anuarovna, Z. D. (2025). Real-time workload prediction and resource optimization for parallel heterogeneous high-performance computing systems architectures. *Ubiquitous Technology Journal*, *1*(1), 40-47.
3. Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An overview on edge computing research. *IEEE access*, *8*, 85714-85728.
4. Cecilia, J. M., Cano, J. C., Morales-García, J., Llanes, A., & Imbernón, B. (2020). Evaluation of clustering algorithms on GPU-based edge computing platforms. *Sensors*, *20*(21), 6335.
5. Debauche, O., Mahmoudi, S., & Guttadauria, A. (2022). A new edge computing architecture for IoT and multimedia data management. *Information*, *13*(2), 89.
6. Galante, G., & da Rosa Righi, R. (2022). Adaptive parallel applications: from shared memory architectures to fog computing (2002–2022). *Cluster Computing*, *25*(6), 4439-4461.
7. Gamatie, A., Devic, G., Sassatelli, G., Bernabovi, S., Naudin, P., & Chapman, M. (2019). Towards energy-efficient heterogeneous multicore architectures for edge computing. *IEEE Access*, *7*, 49474-49491.
8. Gathu, S. (2024). High-performance computing and big data: Emerging trends in advanced computing systems for data-intensive applications. *Journal of Advanced Computing Systems*, *4*(8), 22-35.

Received: August 04, 2025

Volume 38 No. 6s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

9. Hassan, Q. F. (Ed.). (2016). *Innovative research and applications in next-generation high performance computing*. IGI Global.
10. Hwang, K., Dongarra, J., & Fox, G. C. (2013). *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan kaufmann.
11. Lapegna, M., Balzano, W., Meyer, N., & Romano, D. (2021). Clustering algorithms on low-power and high-performance devices for edge computing environments. *Sensors*, *21*(16), 5395.
12. Millett, L. I., & Fuller, S. H. (Eds.). (2011). *The future of computing performance: game over or next level?*. National Academies Press.
13. Mohsin, M. A., Shahrouzi, S. N., & Perera, D. G. (2024). Composing Efficient Computational Models for Real-Time Processing on Next-Generation Edge-Computing Platforms. *IEEE Access*, *12*, 24905-24932.
14. Mzukwa, A. (2024). Exploring Next-Generation Architectures for Advanced Computing Systems: Challenges and Opportunities. *Journal of Advanced Computing Systems*, *4*(6), 9-18.
15. Naayini, P., & Kamatala, S. (2023). High-performance data computing: Parallel frameworks, execution strategies, and real-world deployments. *High-Performance Data Computing: Parallel Frameworks, Execution Strategies, and Real-World Deployments*.
16. Passian, A., & Imam, N. (2019). Nanosystems, edge computing, and the next generation computing systems. *Sensors*, *19*(18), 4048.
17. Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., & Wu, D. O. (2020). Edge computing in industrial internet of things: Architecture, advances and challenges. *IEEE communications surveys & tutorials*, *22*(4), 2462-2488.
18. Reed, N. (2023). Innovative Architectural Designs for Next-Generation HighPerformance Computing. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(3), 1-10.
19. Talati, D. (2021). Decentralized AI: The role of edge intelligence in next-gen computing. *Available at SSRN 5201744*.
20. Usman, S., Mehmood, R., Katib, I., & Albeshri, A. (2022). Data locality in high performance computing, big data, and converged systems: An analysis of the cutting edge and a future system architecture. *Electronics*, *12*(1), 53.
21. Ziya, M. (2022). *Cloud Task Scheduling Dataset*. Kaggle. https://www.kaggle.com/datasets/ziya07/cloud-task-scheduling-dataset
22. Ziya, M. (2022). *IIoT Edge Computing Dataset*. Kaggle. https://www.kaggle.com/datasets/ziya07/iiot-edge-computing-dataset

Received: August 04, 2025