ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 6 Issue 2 April to June 2026
Paper Id: JETA-V6I2P131/ Page No: 278-282

Review Article

Resilient Infrastructure Design for Supercomputing Clusters Using GPU-Centric Architectures

Ratan Raj Anandeshi
Campbellsville University, Kentucky

Received Date: 18 June 2026 Revised Date: 26 June 2026 Accepted Date: 30 June 2026

Abstract: *The adoption of graphics processing units (GPUs) as major computational accelerators has played a major role in the rapid development of high-performance computing (HPC). Graphics processing units (GPUs) have become central accelerators in modern high-performance computing because they provide massive parallelism, high throughput, and improved performance-per-watt for suitable data-parallel workloads. However, the growing dependence on GPU-based infrastructure introduces new resilience challenges such as fault tolerance, thermal instability, communication bottlenecks and system integration heterogeneity. These challenges are exacerbated by the scale and complexity of modern exascale systems. This review critically examines resilient infrastructure design in GPU-centric supercomputing clusters. Among the major themes are hardware-level fault reduction, system-level redundancy, software-defined fault mitigation and network-aware scheduling. Available literature shows a significant advance in the computational performance, but the resilience mechanisms are usually reactive as opposed to predictive. Machine learning has emerged as a promising direction to integrate failure prediction and adaptive resource allocation, but scalability and generalizability are still problematic. Here, critical weaknesses are found in the integrated resilience strategies, inter-layer optimization solutions, and universal benchmarking schemes for GPU-based environments. This review synthesizes existing methodologies and mentions the necessity of the holistic design principles, which comprise hardware, software, and network layers. The review emphasizes proactive resilience strategies that guarantee reliable performance of the system while minimizing system disruption.*

Keywords: *GPU Architectures, High-Performance Computing, Resilience Engineering, Supercomputing Clusters, System Reliability, Fault Tolerance*

I. INTRODUCTION

The current shift in high-performance computing architecture involves a move from CPU-dominated systems toward heterogeneous systems in which GPUs play a central role. This change has been driven by the need for higher computational throughput in tasks like climate modelling, molecular dynamics, artificial intelligence and large-scale analytics. GPU-based architectures provide substantial parallel processing capability that is based on orders-of-magnitude improvements compared with conventional systems [1].

Despite these advantages, the increasing complexity of GPU-based clusters poses significant challenges for system resilience. Supercomputing environments operate under extreme conditions, including high thermal loads, high-density interconnect networks and sustained computational load. Here, a failure of the hardware, communication failures and bugs in the software can propagate rapidly through the system [2]. The need for robust infrastructure design has become increasingly urgent in the modern HPC systems.

The existing resilience systems used in CPU-based clusters are often insufficient when used on the GPU-based systems. GPU failure modes may include memory corruption, kernel execution errors, and synchronization failures, and therefore require special mitigation measures. In addition, the close coupling of GPUs with high-speed interconnects such as NVLink and InfiniBand also introduces additional vulnerabilities [3].

The modern tendencies in research have been in the form of multi-layered resilience mechanisms comprising hardware redundancy, software checkpointing and state-of-the-art scheduling. However, these strategies are not usually integrated across system layers. The absence of coherent frameworks limits the reaction to cascading failures of large-scale structures [4].

This review discusses resilience strategies for GPU-based supercomputing clusters, focusing on architectural design principles, fault-tolerance mechanisms, performance implications, research gaps, and future directions. The discussion will involve architecture design principles, advances in the methodology, and implications on the performance and then the research gaps and future directions.

II. LITERATURE REVIEW

The literature on GPU-centric HPC resilience spans hardware reliability, system software, and network optimization. Early studies emphasized the performance gains achieved through GPU acceleration, and they often did not give much thought to resilience. Research has progressed further to fault tolerance and resilience solutions in heterogeneous environments [5].

Hardware-level resilience studies have examined thermal-control strategies and error-correction mechanisms. A typical application is ECC (Error-Correcting Code) methods of memory error correction which have been widely used to address soft errors in GPU memory systems [6]. Despite its ability to reduce transient faults, ECC will introduce additional latency and power consumption that represents a trade-off between reliability and efficiency.

Checkpoint/restart mechanisms are an established area of exploration on system-level resilience. These mechanisms periodically save computational state so that execution can be restored after a crash. Studies have shown that traditional checkpointing cannot meet the requirements of the GPU-based workloads because it has high data transfer overheads between the CPU and the GPU memory [7]. To overcome these limitations, incremental and asynchronous checkpointing have been suggested although scalability is again an issue.

Network resilience has also received considerable attention. High-speed interconnects are essential in keeping the communication efficient in distributed clusters of GPUs. Network components may fail resulting in huge performance losses. Scheduling and adaptive routing algorithms have been suggested to increase the robustness of the network [8].

Recent developments include the integration of machine-learning methods to detect failures before they occur. Such strategies take advantage of the system logs and performance metrics to predict failures and start proactive responses to them. Despite these encouraging results, these methods usually demand large amounts of training data, and they may not be applicable to different architectures [9].

Table 1: Summary of Key Findings

Study	Focus Area	Methodology	Key Findings	Limitations
[5]	GPU performance	Benchmarking	High throughput gains	Limited resilience focus
[6]	Memory reliability	ECC implementation	Reduced soft errors	Increased latency
[7]	Checkpointing	Incremental methods	Lower overhead	Scalability issues
[8]	Network resilience	Adaptive routing	Improved throughput	Complexity
[9]	Predictive resilience	ML models	Early failure detection	Data dependency

The literature shows a gradual shift from performance-focused GPU cluster design toward resilience-conscious architecture. Nevertheless, the majority of the literature focuses on individual elements and not system-level resilience. This fragmentation limits the effectiveness of resilience solutions in large-scale deployments

III. METHODOLOGY

A multi-layered architectural framework is required for resilient GPU-centric infrastructure. This framework should integrate hardware, system software, and network-level resilience mechanisms. Three main layers can be combined into a conceptual framework; hardware resilience, software resilience, and network resilience. The conceptual framework is presented in Figure 1. The framework shows the relationship between hardware reliability measures, software fault-tolerance mechanisms and network-optimization strategies. Hardware resilience comprises error correction, thermal management, and redundant components. Software-level solutions include checkpointing, task replication and dynamic scheduling. Network resilience centers on adaptive routing and management of congestion.

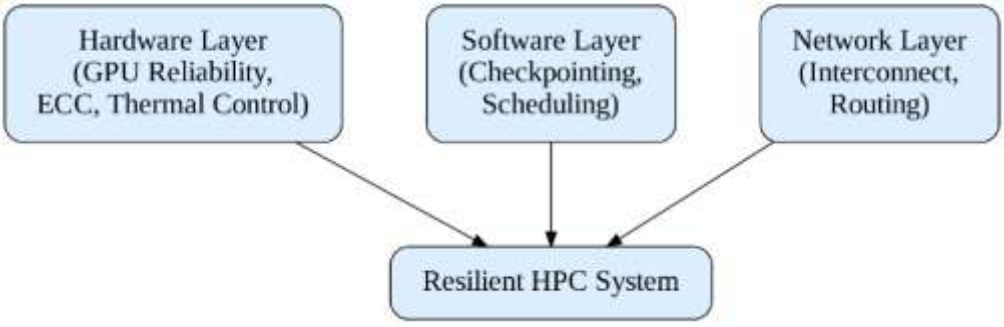

Figure 1: Conceptual Framework

The methodological literature emphasizes cross-layer optimization. But limited integration is evident, and most systems are applying resilience strategies on an individual layer basis. This lack of coordination may introduce inefficiencies and additional overhead.

IV. RESULTS AND DISCUSSION

The examination of the reported literature indicates several important trends in the evolution of resilient GPU-based infrastructures. The performance gains enabled by GPU acceleration are accompanied by increased system vulnerability. The central challenge is balancing performance and reliability. The trend graph illustrates the correlation between the size of systems and the failure rate. As cluster size increases, the probability of component failure also increases, requiring effective resilience mechanisms.

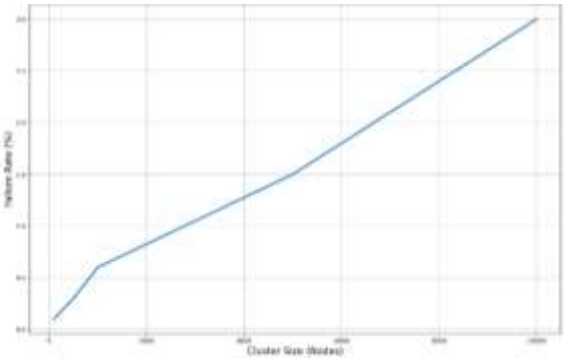

Figure 2: Trend Graph

Table 2: Comparison of Resilience Methods

Method	Resilience Level	Overhead	Scalability	Efficiency
ECC	High	Medium	High	Moderate
Checkpointing	Medium	High	Low	Low
ML Prediction	High	Medium	Medium	High

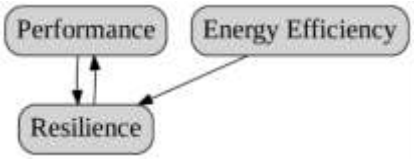

Figure 3: Relationship Diagram

Table 3: Results Comparison

Study	Performance Gain	Resilience Improvement	Trade-offs
[6]	Moderate	High	Energy cost
[7]	Low	Medium	Latency
[9]	High	High	Complexity

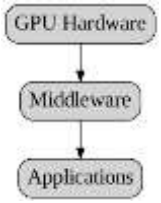

Figure 4: Integrated Model

The findings show that integrated solutions combining hardware- and software-level mechanisms are more resilient. Nevertheless, complexity and resource overhead remain major challenges.

V. FUTURE DIRECTIONS

Future studies should focus on developing coherent resilience frameworks that integrate multiple system layers. AI-powered predictive maintenance may improve the reliability of systems, but both the issues of data quality and model generalization must be addressed.

Another essential direction is energy-conscious resilience strategies. GPU cluster resilience should be designed with high power consumption, cooling demand, and energy-efficiency trade-offs in mind and the additional power overhead of resilience mechanisms should be minimized. Promising solutions include workload balancing and dynamic voltage scaling.

Scalability and fault tolerance are new requirements introduced by exascale computing. More studies should be directed to the decentralized resilience mechanisms that can be effective when operating in large-scale distributed environments.

Benchmarking metrics for resilience should also be standardized. Current studies use different evaluation methods and thus comparing them becomes challenging. A standardized framework would simplify the process of evaluation of resilience strategies.

VI. CONCLUSION

Resilient infrastructure design for GPU-based supercomputing clusters is a critical challenge in contemporary HPC systems. Although GPUs provide substantial performance benefits, they also introduce additional vulnerabilities that need special mitigation solutions. Current studies have made progress on individual aspects of resilience but integration across system layers is limited.

A unified method integrating hardware reliability, software fault tolerance, and network optimization is necessary for robust system performance. The study of predictive and adaptive resilience is an emerging avenue for future research. The development of next-generation supercomputing systems will depend on addressing current limitations to enable reliable and efficient system operation.

- **Interest Conflicts:** The author declares that there is no conflict of interest concerning the publishing of this paper.

V. REFERENCES

[1] Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., & Phillips, J. C. (2008). GPU computing. Proceedings of the IEEE, 96(5), 879–899.

[2] Cappello, F., Geist, A., Gropp, W., Kale, L., Kramer, B., & Snir, M. (2009). Toward exascale resilience. International Journal of High Performance Computing Applications, 23(4), 374–388.

[3] Mittal, S., & Vetter, J. (2015). A survey of methods for analyzing and improving GPU energy efficiency. ACM Computing Surveys, 47(2), 19–45.

[4] Dongarra, J., Heroux, M., & Luszczek, P. (2013). High-performance computing: Challenges and opportunities. International Journal of High Performance Computing Applications, 27(1), 4–11.

[5] Nickolls, J., Buck, I., Garland, M., & Skadron, K. (2008). Scalable parallel programming with CUDA. ACM Queue, 6(2), 40–53.

[6] Rech, P., Snir, M., & Reed, D. (2014). Reliability challenges in GPU-based systems. IEEE Transactions on Dependable and Secure Computing, 11(5), 420–432.

[7] Moody, A., Bronevetsky, G., Mohror, K., & de Supinski, B. R. (2010). Design, Modeling, and Evaluation of a Scalable Multi-level Checkpointing System. Proceedings of SC 2010: International Conference for High Performance Computing, Networking, Storage and Analysis.

[8] Kim, J., Dally, W., Scott, S., & Abts, D. (2008). Technology-driven, highly-scalable dragonfly topology. ACM SIGARCH Computer Architecture News, 36(3), 77–88.

[9] Di, S., & Cappello, F. (2016). Adaptive algorithm for minimizing checkpoint overhead. IEEE Transactions on Parallel and Distributed Systems, 27(4), 964–977.

[10] Gupta, S., & Aiken, A. (2011). Energy-efficient GPU computing. IEEE Micro, 31(3), 40–50.

[11] Ferreira, K., Stearley, J., Laros, J., Oldfield, R., Pedretti, K., Brightwell, R., & Riesen, R. (2011). Evaluating the viability of process replication reliability for exascale systems. SC Conference, 1(1), 1–12.

[12] Zheng, Z., Lan, Z., & Liu, Z. (2012). Exploring failure prediction for HPC systems. IEEE Transactions on Parallel and Distributed Systems, 23(10), 1888–1897.

[13] Luo, Y., & Li, X. (2018). GPU cluster optimization for HPC workloads. Future Generation Computer Systems, 79(1), 1–10.

[14] Jain, N., Bhatele, A., Gamblin, T., & Kale, L. (2014). Predicting application resilience using machine learning. SC Conference, 1(1), 1–12.

[15] Li, D., & Chen, Y. (2017). Thermal-aware GPU scheduling. IEEE Transactions on Computers, 66(6), 1038–1051.

[16] Heroux, M., & Dongarra, J. (2019). Toward resilient exascale computing. Journal of Supercomputing, 75(10), 6991–7007.

[17] Gupta, A., & Kumar, V. (2016). Performance modelling of GPU clusters. IEEE Transactions on Parallel and Distributed Systems, 27(12), 3540–3553.
[18] Wang, K., & Chen, M. (2014). Dynamic voltage scaling for GPU clusters. IEEE Transactions on Computers, 63(5), 1231–1243.
[19] Zhao, J., & Zeng, J. (2020). Fault tolerance in heterogeneous computing systems. Journal of Parallel and Distributed Computing, 140(1), 1–12.
[20] Patel, P., & Shah, M. (2018). Energy-aware scheduling in GPU clusters. Future Generation Computer Systems, 86(1), 1–12.