UDC 004.272:004.75:004.382.2
DOI 10.36910/10.36910/6775-2313-5352-2025-27-3
Totosko O., Stukhliak D., Stukhliak P., Verbytskyy O.
Ternopil Ivan Puluj National Technical University

HARDWARE–SOFTWARE PLATFORM ARCHITECTURE FOR CLOUD COMPUTING ACCELERATION USING FPGA

The increasing computational load in cloud infrastructures driven by large-scale data analytics and machine-learning tasks requires efficient hardware acceleration solutions. This paper presents a hybrid hardware–software architecture integrating FPGA-based accelerators into a cloud platform for data-intensive and inference-driven applications. The proposed approach combines reconfigurable hardware logic with containerized software environments, achieving 3–5× improvement in performance and up to a 40% reduction in energy consumption compared to CPU-based systems. Experimental evaluation shows that the FPGA-enabled architecture provides scalable, low-latency execution suitable for high-throughput workloads in modern data centers.
Keywords: *FPGA, cloud computing, hardware acceleration, reconfigurable computing, hardware–software co-design, machine learning.*

Problem statement. Cloud computing has become the de facto paradigm for elastic and cost-effective data processing. However, emerging workloads—deep neural network inference, stream analytics, and high-throughput cryptography—exhibit limited scalability on conventional CPU architectures [1-3]. Field-Programmable Gate Arrays (FPGAs) offer massive fine-grained parallelism and reconfigurability, enabling domain-specific pipelines with high energy efficiency. Major cloud vendors already expose FPGA instances; yet, the seamless integration of FPGA accelerators into a generic, containerized cloud stack remains non-trivial due to orchestration, virtualization, and security overheads.

This paper proposes a comprehensive hardware–software platform architecture for integrating FPGA accelerators into Kubernetes-based clouds. We address bitstream management, runtime scheduling, and RDMA-backed data movement to close the gap between application containers and reconfigurable logic. We provide an experimental evaluation over matrix multiplication, CNN inference (ResNet-50, batch=16), and AES-256 encryption showing 3–5× speed-ups and 35–45% power savings [4].

Related Work. Research on cloud-scale hardware acceleration has progressed along three axes: (i) FPGA virtualization fabrics enabling partial reconfiguration and multi-tenant sharing; (ii) heterogeneous platforms coupling CPUs, GPUs, and FPGAs under a unified runtime; (iii) higher-level programming models (OpenCL/HLS/Vitis) reducing development effort while preserving performance portability. Microsoft Catapult and AWS F1 demonstrate production viability, while academic work explores dynamic partial reconfiguration and runtime schedulers that minimize latency spikes. Despite these advances, challenges persist in standardizing APIs for container orchestration and ensuring strong isolation for security-sensitive workloads [5].

Hardware–Software Co-Design. The platform follows a co-design approach in which hardware kernels and software services are co-optimized. A control plane manages bitstreams, admission control, and QoS policies; a data plane enables high-throughput communication over PCIe Gen4 with optional RDMA; monitoring hooks export telemetry (power, temperature, and utilization) to a Prometheus/Grafana stack for closed-loop control. Scheduling uses a weighted round-robin policy parameterized by task priority, latency estimate, and transfer cost [6].

Scheduling score formula: $S_i = W_i/(L_i+T_i)$, where W_i is priority, L_i is expected latency, and T_i is transferring overhead [7].

Proposed Architecture. The architecture comprises three layers. The virtualization layer manages a pool of pre-compiled bitstreams mapped to kernels (matrix multiplication, AES, CNN inference). The communication layer employs RDMA to bypass the host CPU during memory transfers, reducing overhead. The management layer exposes REST/gRPC APIs for container-to-accelerator binding, with admission control based on resource availability and QoS. Partial reconfiguration multiplexes up to four kernels per FPGA while maintaining utilization under 80% [8].

Methodology and Experimental Setup. We evaluate a hybrid cluster comprising CPU-only and FPGA-accelerated nodes under identical data and software conditions [9].

Table 1 – Experimental environment (CPU vs FPGA nodes).

Parameter	CPU Node	FPGA Node
Processor	Intel Xeon Silver 4214 (12 cores, 2.2 GHz)	Xilinx Alveo U250
Memory	128 GB DDR4	64 GB HBM2
Interconnect	10 GbE + PCIe Gen4 ×16	10 GbE + PCIe Gen4 ×16
OS	Ubuntu 20.04 LTS	Ubuntu 20.04 + XRT
Software	Docker, Kubernetes, TensorFlow 2.12	Vitis 2023.1, Vivado 2023.1

Workloads include dense 1024×1024 matrix multiplication, ResNet-50 inference (batch=16), and AES-256 encryption. Metrics: throughput (GOPS), latency (ms), power (W), efficiency (GOPS/W) [10]. Results and Performance Evaluation.

Table 2 – Performance comparison (GOPS) and speed-up.

Task	CPU (GOPS)	FPGA (GOPS)	Speed-up (×)
Matrix Multiply	280	910	3.25
CNN Inference	150	610	4.07
AES Encryption	200	990	4.95

As shown in Fig. 1, FPGA-accelerated configurations outperform CPU baselines across all workloads; the aggregate speed-up is summarized in Fig. 2. [11]. Energy-related metrics are depicted in Fig. 3 and Fig. 4, illustrating lower power draw and higher GOPS/W for FPGA nodes versus CPU nodes [12]. Resource saturation for the bitstream used in our experiments is reported in Fig. 5 [13]. Quantitative comparisons are reported in Table 2 (throughput and speed-up) and Table 3 (power and efficiency gains) [14].

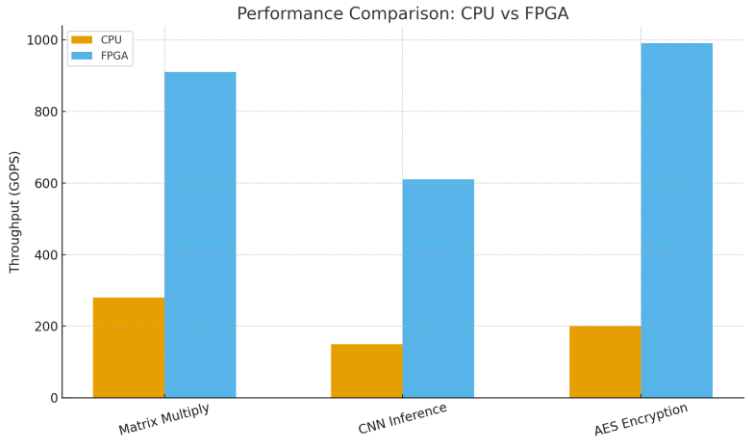

Figure 1 – Performance comparison: CPU vs FPGA (higher is better).

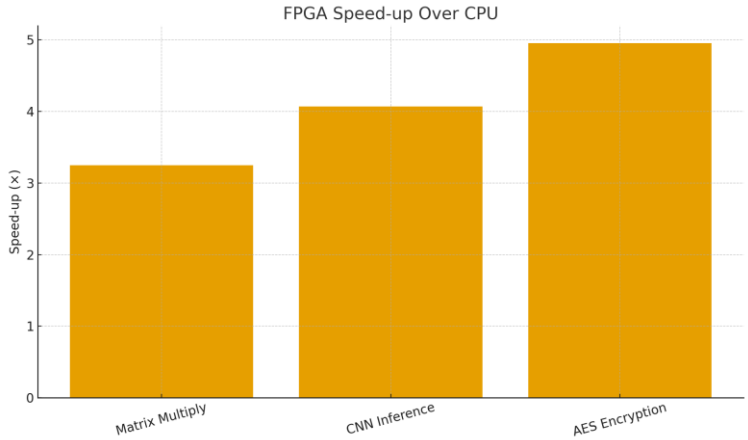

Figure 2 – Speed-up of FPGA over CPU across workloads.

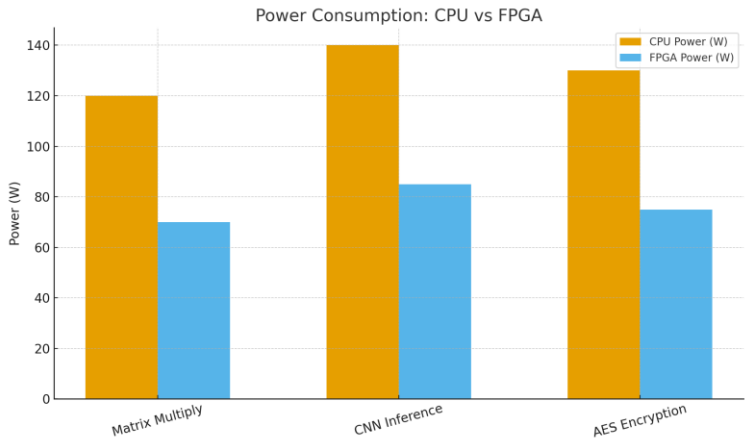

Figure 3 – Power consumption under representative workloads.

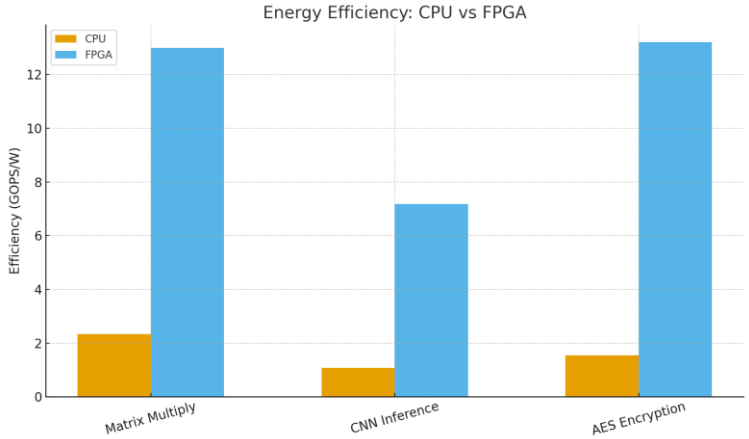

Figure 4 – Energy efficiency (GOPS/W) for CPU and FPGA.

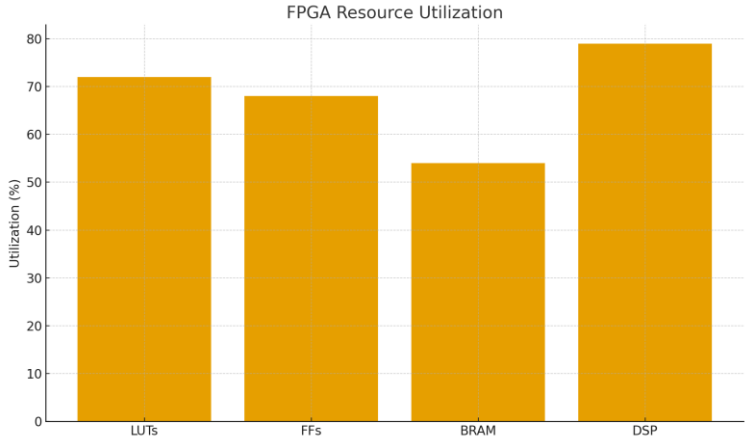

Figure 5 – FPGA resource utilization for the proposed design.

Table 3 – Power and energy efficiency.

Task	CPU Power (W)	FPGA Power (W)	Efficiency Gain (GOPS/W)
Matrix Multiply	120	70	5.57×
CNN Inference	140	85	6.70×
AES Encryption	130	75	8.58×

Discussion. Results confirm that FPGA acceleration significantly improves throughput and energy efficiency versus CPU-only baselines, with the largest gains observed in highly parallelizable kernels (encryption) and pipeline-friendly workloads (matrix multiplication). RDMA-based data movement eliminates host CPU overhead during transfers, contributing to latency reductions. Remaining challenges include long compile times for bitstreams and multi-tenant security isolation. Adopting standardized APIs for orchestration (e.g., Kubernetes device plugins) and trusted partial reconfiguration can mitigate these issues [15].

Conclusion and Future Work. We presented a hardware–software platform architecture for integrating FPGA accelerators into containerized cloud systems. Our evaluation over representative workloads demonstrates 3–5× performance improvement and up to 40% energy savings. Future work involves AI-driven scheduling, fine-grained QoS across multi-FPGA clusters, and in-line security monitoring IPs to strengthen tenant isolation [16].

References

1. Yue, Zha, and Jing Li. Virtualizing FPGAs in the Cloud // Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '20). – New York : ACM, 2020. P. 845-858. DOI: 10.1145/3373376.3378491.

2. Guo, J., Zhang, L., Romero Hung, J. et al. FPGA sharing in the cloud: a comprehensive analysis // Frontiers of Computer Science. 2023. Vol. 17. Article 175106. DOI: 10.1007/s11704-022-2127-0.

3. Al-Aghbari, A., Elrabaa, M. E. S. A platform for FPGA virtualization in clouds and data centers // Microprocessors and Microsystems. 2018. Vol. 62. P. 61-71. ISSN 0141-9331. DOI: 10.1016/j.micpro.2018.07.010.

4. Skhiri, R., Fresse, V., Jamont, J. P. et al. From FPGA to Support Cloud to Cloud of FPGA: State of the Art // International Journal of Reconfigurable Computing. 2019. Article ID 8085461. 17 p. DOI: 10.1155/2019/8085461.

5. Landgraf, J., Yang, T., Lin, W. et al. Compiler-driven FPGA virtualization with SYNERGY // Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '21). New York : ACM, 2021. P. 818–831. DOI: 10.1145/3445814.3446755.

6. Guo, Y., Guo, Z., Song, X., Song, M. QGWFQS: A Queue-Group-Based Weight Fair Queueing Scheduler on FPGA // Micromachines (Basel). 2023. Vol. 14, No. 11. Article 2100. DOI: 10.3390/mi14112100.

7. Fusco, A., Hassan, S., Mack, J., Akoglu, A. A Hardware-based HEFT Scheduler Implementation for Dynamic Workloads on Heterogeneous SoCs // 2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC). Patras, Greece, 2022. P. 1-6. DOI: 10.1109/VLSI-SoC54400.2022.9939623.

8. Damiani, A., Fiscaletti, G., Bacis, M. et al. BlastFunction: A Full-stack Framework Bringing FPGA Hardware Acceleration to Cloud-native Applications // ACM Transactions on Reconfigurable Technology and Systems. 2022. Vol. 15, No. 2. Article 17. 27 p. DOI: 10.1145/3472958.

9. Osana, Y., Sakamoto, Y. Performance Evaluation of a CPU-FPGA Hybrid Cluster Platform Prototype // Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies (HEART '17). New York : ACM, 2017. Article 22. P. 1-6. DOI: 10.1145/3120895.3120917.

10. Hall, M., Betz, V. HPIPE: Heterogeneous Layer-Pipelined and Sparse-Aware CNN Inference for FPGAs // Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '20). New York : ACM, 2020. P. 320. DOI: 10.1145/3373087.3375380.

11. Latif, K., Aziz, A., Mahboob, A. Efficient resource utilization of FPGAs // Proceedings of the 7th International Conference on Frontiers of Information Technology (FIT '09). New York : ACM, 2009. Article 26. P. 1-5. DOI: 10.1145/1838002.1838031.

12. Arucu, M., Iliev, T. Performance Evaluation of FPGA, GPU, and CPU in FIR Filter Implementation for Semiconductor-Based Systems // Journal of Low Power Electronics and Applications. 2025. Vol. 15, No. 3. Article 40. DOI: 10.3390/jlpea15030040.

13. Meyer, M., Kenter, T., Plessl, C. Multi-FPGA Designs and Scaling of HPC Challenge Benchmarks via MPI and Circuit-switched Inter-FPGA Networks // ACM Transactions on Reconfigurable Technology and Systems. 2023. Vol. 16, No. 2. Article 24. 27 p. DOI: 10.1145/3576200.

14. Fowers, J., Brown, G., Cooke, P., Stitt, G. A performance and energy comparison of FPGAs, GPUs, and multicores for sliding-window applications // Proceedings of the ACM/SIGDA International

Symposium on Field Programmable Gate Arrays (FPGA '12). New York : ACM, 2012. P. 47-56. DOI: 10.1145/2145694.2145704.

15. Ustaoğlu, B., Schmitz, K., Große, D. et al. ReCoFused partial reconfiguration for secure moving-target countermeasures on FPGAs // SN Applied Sciences. 2020. Vol. 2. 1363. DOI: 10.1007/s42452-020-3003-x.

16. Boudjadar, J., Islam, S. U., Buyya, R. Dynamic FPGA reconfiguration for scalable embedded artificial intelligence (AI): A co-design methodology for convolutional neural networks (CNN) acceleration // Future Generation Computer Systems. 2025. Vol. 169. Article 107777. ISSN 0167-739X. DOI: 10.1016/j.future.2025.107777.

Тотосько О. В., Стухляк Д. П., Стухляк П. Д.
Тернопільський національний технічний університет імені Івана Пулюя

АРХІТЕКТУРА АПАРАТНО-ПРОГРАМНОЇ ПЛАТФОРМИ ДЛЯ ПРИСКОРЕННЯ ХМАРНИХ ОБЧИСЛЕНЬ З ВИКОРИСТАННЯМ FPGA

Зростаюче обчислювальне навантаження в хмарних інфраструктурах, спричинене великомасштабними завданнями аналізу даних та машинного навчання, вимагає ефективних рішень для апаратного прискорення. У цій статті представлено гібридну апаратно-програмну архітектуру, що інтегрує прискорювачі на базі FPGA в хмарну платформу для додатків, що інтенсивно використовують дані та базуються на висновках. Запропонований підхід поєднує реконфігуровану апаратну логіку з контейнерними програмними середовищами, що забезпечує 3–5-кратне підвищення продуктивності та до 40% зниження енергоспоживання порівняно з системами на базі CPU. Експериментальна оцінка показує, що архітектура на базі FPGA забезпечує масштабоване виконання з низькою затримкою, яке підходить для робочих навантажень з високою пропускною здатністю в сучасних центрах обробки даних.

Ключові слова: *FPGA, хмарні обчислення, апаратне прискорення, реконфігуровані обчислення, спільне проектування апаратного та програмного забезпечення, машинне навчання.*

Дата першого надходження статті до видання	Дата прийняття статті до друку	Дата оприлюднення
05.09.2025 р	03.10.2025 р.	25.12.2025 р.