***Bohdan Boretskyi,*** *student,*

*Lviv Polytechnic National University*

*ORCID: 0009-0009-5295-5558*

***Supervisor: Halyna Vlakh-Vyhrynovska****, associate professor, PhD,*

*Lviv Polytechnic National University*

*ORCID: 0000-0003-4429-1578*

**LOW-COST DISTRIBUTED COMPUTING WITH APACHE SPARK: A REVIEW OF EDUCATIONAL AND EXPERIMENTAL RASPBERRY PI CLUSTERS**

Distributed computing is crucial in handling large data volumes and performing complex algorithms in a wide range of fields, from scientific modeling to real-time analytics. Distributed systems, in contrast to centralized ones, split up computations across numerous nodes, supporting parallel execution as well as fault tolerance. Apache Spark has become one of the most popular frameworks for such operations because of its in-memory model for processing, fault tolerance by means of Resilient Distributed Datasets (RDDs), and iterative operations support.

Spark is based on a master-worker architecture, in which a central node (driver) assigns tasks to numerous executor nodes. It is deployable in a variety of environments, such as standalone, YARN, Mesos, or Kubernetes, and presents a modular collection of libraries for SQL query, graph processing, machine learning, and stream data processing. Support for executing Spark in ARM-based systems has also opened up new routes for low-cost experimentation, as well as in-class deployment, with the increasing performance potential of single-board computers like the Raspberry Pi.

Recent research has demonstrated the potential of using Raspberry Pi clusters to simulate distributed environments for both research and educational purposes. One study utilized a 64-node Raspberry Pi cluster to perform topology optimization tasks, emphasizing energy efficiency and accessibility [1]. Another investigation explored a hybrid environment combining Raspberry Pi nodes with Google Colab to teach parallel and heterogeneous computing, highlighting the benefits of hands-on experience with distributed workflows [2]. Additional findings reported the successful implementation of a modular, portable instructional system using pre-configured Raspberry Pi clusters, which led to significant improvements in student understanding of parallel computing concepts [3].

Beyond academic applications, Raspberry Pi clusters have also been evaluated in terms of energy efficiency and cost-effectiveness. An experiment involving a 25-node cluster assessed its performance and power consumption, demonstrating its viability as an eco-friendly solution for lightweight computing tasks, serving as an alternative to traditional cloud-based setups [4].

One of the most attractive features of such physical cluster installations is the visibility and control these allow. In contrast with proprietary, commercial clouds, local clusters offer direct access to the network layer, system configuration, and hardware interfaces, particularly valuable within the academic community. Furthermore, through the use of ARM-compatible software stacks and open-source orchestration software, users are able to recreate real distributed systems within budget limitations.

Though the scale of Raspberry Pi clusters is subject to limits when working with production-level workloads, their pedagogical utility is well-documented. They present a physical, hackable platform within which distributed algorithms can be explored, fault tolerance approaches can be tested, and observations can be made of resource scheduling — all within the unabstracted, low-cost environment of commercial clouds. As distributed computations become more central within modern software systems, such low-cost platforms are a valuable tool in bridging the divide between academic instruction and practical comprehension.

In parallel, advances in low-energy distributed hardware as well as orchestration software have broadened Raspberry Pi cluster applicability. Improved single-board computers offer better RAM as well as the latest ARM-based processors in a 64-bit configuration, allowing more intensive workloads while not taking a substantial hit in terms of power consumption. Lightweight orchestration software such as K3s has been successfully used within Raspberry Pi clusters, supporting automated scheduling and scaling of distributed, container-based code across constrained devices [5]. Further, hybrid edge-cloud approaches are gaining prominence, in which local clusters execute latency-insensitive computations while delegating computationally intensive tasks outside. This design minimizes network dependency as well as response time in data-intensive uses. Experimental results indicate that such hybrid designs can achieve performance similar to centralized ones under certain workloads, while enjoying the advantages of local control as well as lower operational expenses [6].

**References**

1. Zhang, Z.-D., Yu, D.-Y., Ibhadode, O., et al. (2025). TopADDPi: An Affordable and Sustainable Raspberry Pi Cluster for Parallel-Computing Topology Optimization. Processes, 13(3), 633. https://doi.org/10.3390/pr13030633
2. Xu, Z. (2023). Teaching Heterogeneous and Parallel Computing with Google Colab and Raspberry Pi Clusters. Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis, 308–313. https://doi.org/10.1145/3624062.3624095
3. Shoop, E., Matthews, S. J., Brown, R., & Adams, J. C. (2025). Hands-on Parallel & Distributed Computing with Raspberry Pi Devices and Clusters. Journal of Parallel and Distributed Computing, 104996. https://doi.org/10.1016/j.jpdc.2024.104996
4. McDonald, J. (2020). Evaluation of Power and Performance in Raspberry Pi 2 Cluster for Educational Use. Electronics, 5(4), 61. https://doi.org/10.3390/electronics5040061
5. Čilić, I., Krivić, P., Podnar Žarko, I., & Kušek, M. (2023). Performance Evaluation of Container Orchestration Tools in Edge Computing Environments. Sensors, 23(8), 4008. https://doi.org/10.3390/s23084008
6. Shwe, T., & Aritsugi, M. (2024). Optimizing Data Processing: A Comparative Study of Big Data Platforms in Edge, Fog, and Cloud Layers. Applied Sciences, 14(1), 452. https://doi.org/10.3390/app14010452