A Hardware/Software Co-Design Approach for Versal-Based K-means Acceleration

Eleonora Cabai, Giuseppe Sorrentino, Marco Domenico Santambrogio, Davide Conficconi
Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano
eleonora.cabai@mail.polimi.it, {giuseppe.sorrentino, marco.santambrogio, davide.conficconi}@polimi.it

***Abstract*—K-means is a pivotal clustering algorithm widely accelerated through specific architectures to mitigate the computational load. Nevertheless, existing solutions lack a comprehensive co-design approach for optimization. By combining AI engines with the reconfigurability of FPGAs, this research focuses on Versal systems and proposes a co-design process that addresses both software algorithm optimization and hardware bottleneck analysis. The proposed accelerator achieves a speedup of up to $21.41\times$ compared to state-of-the-art solutions.**

***Index Terms*—Co-Design, AI Engine, Versal, FPGA, K-means**

I. Introduction

Clustering is a widely employed unsupervised learning technique [1], and K-means is one of the most studied algorithms, with Lloyd's version enhancing performance [2] while MacQueen's sacrifices part of them to save accuracy [3]. Despite advancements in hardware-accelerated K-means, few solutions target heterogeneous systems, missing a clear co-design process. Consequently, emerging architectures such as Versal systems are underexplored. Therefore, this work aims at analyzing the existing K-means algorithms while **proposing a hardware-software co-design approach** to accelerate the computation over Versal's heterogeneous layers.

II. Methodology

We first target MacQueen's version to have both performance and accuracy and partition the computation step on the AI Engine (AIE) to leverage their data parallel features while using the FPGA for data movement.

MacQueen algorithm introduces severe data dependencies, which limits our parallel capabilities. Indeed, a synchronization step is needed to update the cluster coordinates at each new incoming point. Considering a workload parallelization strategy that splits the points across multiple compute units designed on the AIEs, the synchronization requires sharing information among the AIE about the newly updated clusters, as no shared memory exists between the AIEs.

Thus, we pivot to Lloyd's version, which postpones the synchronization phase and eliminates all the data dependencies between the computations of each pair of points, enhancing parallelization capabilities. Thanks to this software choice, we can scale the design, increasing the number of AIE dedicated to the computation while using the Field Programmable Gate Array (FPGA) kernel for moving data. As AIEs are statically scheduled processors, the FPGA must compute and forward precise information about the number of operations to execute each AIE tile. Thanks to the FPGA flexibility in computing such parameters and spreading information, we could parallelize the design, leveraging the spatial parallel capabilities of novel heterogeneous Versal architectures.

III. Results

We relied on HLS 2022.2 and Vitis for the design, while the targeted a VCK5000 coupled with an AMD EPYC 7V13.

Comparing AIE versions of $MacQueen$ and $Lloyd$, the latter demonstrates an average speedup of $2.48\times$ over the former when considering single-AIE implementations while attaining up to $15.81\times$ speedup in its fully optimized version with 32 AIEs. We remark that performance improvements grow with larger input sizes for more effective workload distributions. Then we compare our $32-$ AIE $Lloyd$ accelerator's against FAISS [4] and Scikit-Learn [5] LLoyd's algorithms, accelerated on mid-to-high-end architectures (NVIDIA RTX4070 and A5000 GPUs, and EPYC 7V13 CPUs). We attain high speedup against CPU versions, with a peak of $21.41\times$ against Scikit-Learn, while being comparable to the GPU versions.

IV. Conclusions & Acknowledgement

We propose a co-design process for Versal-based K-means acceleration, parallelizing the workload across multiple AIEs and FPGA kernels achieving remarkable speedup.

Acknowledgment - This work has financial support from ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU. This work was supported in part by AMD under the Heterogeneous Accelerated Compute Clusters (HACC) program. The authors are grateful to AMD University Program support.

References

[1] J. Bhimani, M. Leeser, and N. Mi, "Accelerating k-means clustering with parallel implementations and gpu computing," in *2015 IEEE High Performance Extreme Computing Conference (HPEC)*, 2015, pp. 1–6.

[2] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967. [Online]. Available: https://api.semanticscholar.org/CorpusID:6278891

[4] FAISS, "FAISS Official Website," 2025. [Online]. Available: https://faiss.ai/index.html

[5] Scikit-learn Developers, "Clustering — Scikit-learn Documentation," 2025. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html