

Active Compute Memory: Enhancing Memory and Processing in Near-Memory Architectures for Vector Classification

Victor Xirau*, Pouya Esmaili*, Petar Radojković*

*Barcelona Supercomputing Center, Barcelona, Spain

E-mail: {victor.xirau, pouya.esmaili, petar.radojkovic}@bsc.es

Keywords—Computer Architecture, Active Compute Memory ACM, Processing In Memory PIM, Classification in Memory

I. EXTENDED ABSTRACT

This study evaluates the Active Compute Memory (ACM) architecture [1] for vector classification, diverging from its original use while maintaining its microarchitecture. Conducted in collaboration with La Salle Barcelona University for a final thesis, we combined analytical modeling and hardware simulation to validate our findings. We found that ACM excels in speed and achieves up to 95% accuracy in tasks with 2-3 classes. However, accuracy drops below 50% for tasks with more classes, indicating a need for further optimization for complex classifications. These results reveal a trade-off between speed and accuracy, showcasing ACM's potential as an alternative to traditional CPU-based methods for data-intensive tasks, and contributing to computer architecture advancements with practical implications for real-world applications.

A. Enhancing ACM with Vector Classification

Introduced by BSC, the ACM architecture innovates in-memory computing by efficiently sorting key-value pairs within DRAM [1]. This system, integrating Data-RAM, Sort-RAM (with KeyTable, MetaTable, and Control Logic), significantly surpasses traditional CPU-based methods in both performance and energy efficiency.

a) ACM's Original Functionality and Limitations:

Initially, ACM was crafted for sorting, allowing data retrieval by sorted keys via indirect addressing through KeyTable (KT) and MetaTable (MT) manipulation. However, its basic classification algorithm—categorizing data by labels—struggles with the complexity and variety of contemporary datasets that often feature unlabeled, multidimensional data.

- *Methodological Shift:* The move towards vector classification entails a transition from processing elements with explicit labels to classifying multidimensional vectors against expected class vectors.
- *Implementation Considerations:* The vector classification process within ACM leverages the existing architecture, repurposing the MT to store expected vectors for each class and utilizing the Control Logic and Bots for dynamic vector comparison and classification. This method effectively transforms the ACM from a sorting device into a powerful classification tool.

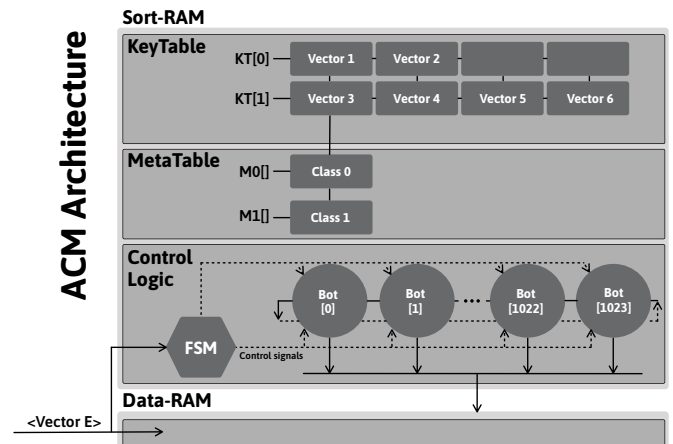


Fig. 1. Adaptation of ACM architecture for vector classification.

b) Exploration of Advanced Classification Algorithms:

The ACM's reliance on exact match comparisons was challenged by its limitations in handling complex, multidimensional data. To address this, a study explored alternative algorithms to improve accuracy. The Absolute Difference per Element Algorithm stood out, assessing the absolute difference between input vectors and expected class vectors against a set threshold, typically 0.75, for a more nuanced classification. This approach demonstrated superior flexibility and accuracy over other algorithms, confirmed by tests on diverse datasets like MNIST [2], Breast Cancer Wisconsin [3], Iris [4], Titanic [5], and Wine [6] and replicating the ACM's behaviour in CPU to run them.

c) *Hardware Considerations for Implementation:* Implementing the Absolute Difference per Element Algorithm within the ACM architecture required minimal additional hardware overhead. The proposed design utilizes comparators and multiplexers to compute absolute differences, with BOT units performing subtraction operations.

B. Experimental Environment

The evaluation of ACM Vector Classification within this study adopts an Analytical Model approach, this strategy is essential for delving into the performance characteristics and efficiency of the ACM algorithms, providing a detailed examination in the absence of the actual hardware. The move towards an Analytical Model stems from the necessity to simulate

and assess algorithmic behaviors and potential optimizations realistically, circumventing the limitations inherent in purely theoretical computational complexity analyses.

To bridge the gap between theoretical analysis and tangible hardware evaluation, we enhanced the publicly available ZSim [7] and DRAMSim3 [8] simulators. These enhancements are tailored to accommodate the specific requirements of ACM evaluation, enabling practical benchmarks and performance validation of the ACM architecture. The utilization of the ZSim simulator, in particular, has been crucial for corroborating the findings of the Analytical Model, ensuring that the ACM's conceptual design does not introduce computational bottlenecks.

C. Results

The ACM architecture's exploration in classification tasks across different datasets required custom processing for each, accommodating up to 120 elements per vector within class and dimension constraints. Through CPU emulation, variances in ACM's performance were observed, particularly on the MNIST dataset where accuracy ranged from about 10% to 19%, significantly below state-of-the-art (SOTA) methods, underscoring the algorithm's limitations and the impact of datasets with high zero-value prevalence.

Execution speeds were notably quick, mostly under 0.2 seconds, except for larger datasets like MNIST. This demonstrates ACM's fast processing but highlights a trade-off with accuracy. When compared to conventional CPU-based approaches, the ACM showed reduced execution times with about 80% accuracy in simpler scenarios (2-3 classes) but saw a decrease to below 50% in more complex classifications, indicating a pressing need for algorithmic improvement and optimization to enhance its classification performance across a broader range of applications.

D. Conclusion

This study elucidates the potential and challenges of employing ACM for vector classification tasks. While the ACM exhibits exceptional speed, particularly in simpler classification scenarios with fewer output classes, the accuracy in more complex applications necessitates improvement. The findings from real-world dataset applications reveal a critical trade-off between execution speed and classification accuracy. Moving forward, enhancing the ACM's algorithmic framework to better

accommodate a broader spectrum of classification tasks without compromising on speed or accuracy remains a pivotal area of research.

II. ACKNOWLEDGMENT

This thesis has been formally recognized and accepted by La Salle University, where it was awarded the distinction of Honors. While this work has not yet been published, efforts are underway to prepare a manuscript for submission.

REFERENCES

- [1] P. Esmaili-Dokht, M. Guiot, P. Radojković, X. Martorell, E. Ayguadé, J. Labarta, J. Adlard, P. Amato, and M. Sforzin, "O (n) key-value sort with active compute memory," *IEEE Transactions on Computers*, 2024.
- [2] Ultralytics, "Mnist," n.d. [Online]. Available: <https://docs.ultralytics.com/datasets/classify/mnist>
- [3] "Breast cancer wisconsin (diagnostic) data set," n.d. [Online]. Available: <https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>
- [4] panData, "Unveiling the mysteries of the iris dataset: A comprehensive analysis and machine learning," March 2023. [Online]. Available: <https://levelup.gitconnected.com/unveiling-the-mysteries-of-the-iris-dataset-a-comprehensive-analysis-and-machine-learning-f5c4f9dbcd6d>
- [5] N. Donges, "Predicting the survival of titanic passengers," May 2018. [Online]. Available: <https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8>
- [6] "Uci machine learning repository," n.d. [Online]. Available: <https://archive.ics.uci.edu/dataset/109/wine>
- [7] D. Sanchez and C. Kozyrakis, "Zsim: fast and accurate microarchitectural simulation of thousand-core systems," in *40th Annual International Symposium on Computer Architecture (ISCA)*, 2013.
- [8] S. Li, Z. Yang, D. Reddy, A. Srivastava, and B. Jacob, "Dramsim3: A cycle-accurate, thermal-capable dram simulator," *IEEE Computer Architecture Letters*, vol. 19, no. 2, 2020.



Victor Xirau received his double BSc degree in Computer Engineering and Multimedia Engineering from La Salle Barcelona University in 2023. He was a Teacher Assistant for Operating Systems and Compiler Design from 2021 to 2023. He then went on to complete his final degree's thesis with the Memory Systems group at the Barcelona Supercomputing Center (BSC) in 2022. Since 2023, Victor is a full-time Research Engineer in the Memory Team at BSC and serves as a part-time Lecturer at La Salle Barcelona.