Project (deadline mid June 2025)

The goal is to investigate the optimal cache architecture (you already did this for the aes and tinyjpeg benchmarks) and the virtual memory effects.
In addition to the aes and tinyjpeg benchmarks, examine additional bechmarks, such as as the following:

i) Light gradient boosting ML benchmark – LGB (an example benchmark, eclass)

ii) A single thread uses the least squares method to compute the trend of 1000 random-generated values (representing temparature), and prints the prediction.
(example code): https://stackoverflow.com/questions/5083465/fast-efficient-least-squares-fit-algorithm-in-c
Extend this for 2, 4, 8 threads.

To explore the behavior of these benchmarks on a system with virtual memory, you may use various virtual memory simulators, which are open-source for educational and research purposes, such as the following:
https://github.com/juliorodrigues07/virtual_memory_sim.git

Methodology
Step 1. run the virtual memory simulator and gather results as in the report included, in order to understand simulation of virtual memory

Step 2. use Pin tools to collect memory traces from aes, tinyjpeg, LGB, least-squares
You may use open source code, e.g.,
(a)https://reverseengineering.stackexchange.com/questions/20289/how-to-get-per-function-memory-accesses-using-pin-tool
(b)https://gist.github.com/sudhackar/b3c355ff7f39c2a30c19c88053b60939

Step 3. Plot the memory traces behavior, the cache behavior for LGB and least-squares and the virtual memory behavior for aes, tinyjpeg, LGB, least-squares.

Deliverables: code/scripts, report (screenshots, graphs), conclusions.

Deadline: 14 June 2026