Supplementary Material

Beyond Big-O: Why Optimizing Data Structure is Futile for Exact Streaming Top-K on IID Data

About

- Description: The source code of experiment evaluation
- Authors: Pengfei Wang, Xiaolei Guo, Xian Wu
- Date: December 14, 2025

Source Code Files

- topk_component.cpp: The core data structures (TFA and MinHeap)
- topk_algorithm.cpp: The full algorithms (TFA-Topk and MinHeap-Topk)
- test_dataset.cpp: The IID data generator
- test_end2end.cpp: The end to end test cases
- test_end2end_hardware.cpp: The hardware performance counter of end-to-end algorithms
- test_microbenchmark.cpp: The micro-benchmark test cases
- test_microbenchmark_hardware.cpp: The hardware performance counter of core data structures

Experiment Environment Setup

- Compiler: GNU g++ 11.4.0
- OS: Unbutu 22.04 LTS (Linux Kernel 6.6)
- CPU: AMD Ryzen 9 PRO 7940HS (Zen 4) 4.00 GHz, 8MB L2 Cache, 16MB L3 Cache
- Memory: 16G DDR4
- Profiler: Linux perf (version 5.15)
- Language: C++ Standard 20

Compilation Instructions

- $ g++ -std=c++20 -O3 ./test_end2end.cpp -o end2end
- $ g++ -std=c++20 -O3 ./test_end2end_hardware.cpp -o end2end_hardware
- $ g++ -std=c++20 -O3 ./test_microbenchmark.cpp -o microbenchmark
- $ g++ -std=c++20 -O3 ./test_microbenchmark_hardware.cpp -o microbenchmark_hardware

Executing Test Cases

- Micro-Benchmark Test:
  - $ ./microbenchmark 100000 10 #(N = 100000, k = 10)
  - $ ./microbenchmark_hardware 100000 10 1 #(N = 100000, k = 10, data structure = TFA)
  - $ ./microbenchmark_hardware 100000 10 2 #(N = 100000, k = 10, data structure = MinHeap)
- End-to-End Test:
  - $ ./end2end 100000 10 #(N = 100000, k = 10)
  - $ ./end2end_hardware 100000 10 1 #(N = 100000, k = 10, algorithm = TFA-Topk)
  - $ ./end2end_hardware 100000 10 2 #(N = 100000, k = 10, algorithm = MinHeap-Topk)