

Performance Analysis and Tuning on Modern CPUs (second edition)

First edition: https://book.easyparf.net/perf_book.

Github repo: <https://github.com/dendibakh/perf-book>.

Below is the list of planned updates for the second edition. I need help with some of these topics. Let me know (dendibakh@gmail.com) if you possess knowledge on a particular topic and would like to contribute! Also, feel free to leave your comments.

The planned new table of contents is here (work in progress):

https://github.com/dendibakh/perf-book/blob/main/new_toc.md

Big things to add:

- [help appreciated] performance considerations on x86, ARM, and RISC-V (similarities and differences).
- [need help] performance monitoring on non-Intel platforms (AMD uProf, Apple Instruments)
- [need help] eBPF for performance analysis.
- questions and exercises at the end of each chapter.
- a section on current and future trends in HW and SW performance.

Things to add:

- optimizing IO
- a brief overview of tools for performance analysis on Mac and Windows.
- details of some low-level perf-related events (memory order violation, 4K aliasing)
- explain how power-saving features affect performance (C-states, P-states, core parking, etc). Also affinity and task scheduling.
- binary translation (Rosetta).
- microarchitecture-specific optimizations (instruction latencies and throughput at uops.info, memory order violation, 4K aliasing, etc.)
- simulators (uica, gem5, sniper) and how they can be useful in performance analysis (mainly uica).
- code optimization guidelines. <https://www.codee.com/knowledge/>
- [maybe] performance considerations for languages other than C++ (mainly Java, Rust, maybe Zig).

To improve:

- section on multithreaded applications
- section on SIMD and programming with intrinsics
- section on huge pages, page walks, etc
- add unroll & jam in the section with loop optimizations.
- shave off some low-value footnotes

This list will likely grow bigger 😊