

193 Branches

0 Tags

Go to file

About file

Code

...

npt-1707

Fix potential vulnerabl...

f49680e · 5 days ago

.github	Reverted chan...	8 months ago
buildbot	[CI] catch error...	2 months ago
img	Downsized you...	3 years ago
labs	Fix potential vu...	5 days ago
tools	[NewLab] com...	2 months ago
.editorc...	Added misc files	4 years ago
.gitignore	Update gitignor...	2 months ago
Contribu...	Updated contri...	4 years ago
GetStart...	Updated OS v...	last year
Quickst...	Migrated to Cla...	2 years ago
Quickst...	Migrated to Cla...	2 years ago
Quickst...	Migrated to Cla...	2 years ago
READM...	Update year 2...	2 months ago
logo.jpg	Added misc files	4 years ago

This is an online course where you can learn and master the skill of low-level performance analysis and tuning.

- Readme
- Activity
- 3.1k stars
- 106 watching
- 300 forks

Report repository

Releases

No releases published

Sponsor this project

dendibakh Denis Bakhva...

patreon.com/dendibakh

https://paypal.me/DenisBakhvalov

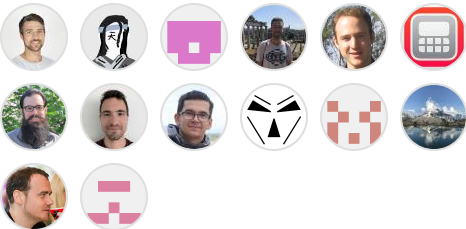
Learn more about GitHub Sponsors

Packages

No packages published

README

Contributors 16





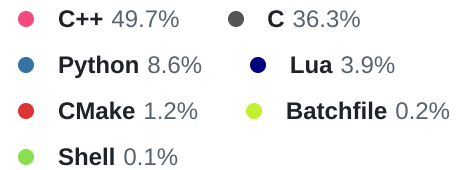
 Benchmark on Linux + Alderlake **passing**

 Benchmark on Mac OS + M1 **passing**

 Benchmark on Windows + Zen3 **passing**

 Benchmark on Linux + Coffeelake **passing**

## Languages



# Performance Ninja Class

 Subscribers **2k**  Follow @dendibakh  Stars **3.1k**

This is an online course where you can learn to find and fix low-level performance issues, for example CPU cache misses and branch mispredictions. It's all about practice. So we offer you this course in a form of lab assignments and youtube videos. You will spend at least 90% of the time analyzing performance of the code and trying to improve it.



Each lab assignment focuses on a specific performance problem and can take anywhere from 30 mins up to 4 hours depending on your background and the complexity of the lab assignment itself. Once you're done improving the code, you can submit your solution to Github for automated benchmarking and verification.

Performance Ninja is supported on Linux, Windows, and Mac, and is run on all the recent HW including Intel's 12th-gen Alderlake, AMD's Zen3, and Apple's M1 CPUs. You can observe the effect of your optimizations on a variety of modern platforms.

Prerequisites: basic C++ skills are an absolute must-have for the course. Denis' [book](#) "Performance Analysis and Tuning on Modern CPUs" is recommended as an introduction to performance analysis basics. Knowledge of compilers, computer architecture, and the ability to read assembly code is a plus.

Before you start working on lab assignments, make sure you read [Get Started page](#) and watch the [warmup video](#).

Lab assignments in this project are implemented in C++. Also, Performance Ninja was ported to:

- Rust ([perf-ninja-rs](#)), thanks to @grahamking.
- Zig ([perf-ninja-zig](#)), thanks to @JonathanHallstrom.

## Lab assignments

---

- Core Bound:
  - [Vectorization 1](#)
  - [Vectorization 2](#)
  - [Function Inlining](#)
  - [Dependency Chains 1](#)
  - [Dependency Chains 2](#)
  - [Compiler Intrinsics 1](#)
  - [Compiler Intrinsics 2](#)
  - [Compiler Intrinsics 3](#)
  - [Compiler Intrinsics 4](#)
- Memory Bound:
  - [Data Packing](#)
  - [Loop Interchange 1](#)
  - [Loop Interchange 2](#)
  - [Loop Tiling](#)
  - [SW memory prefetching](#)
  - [False Sharing](#)
  - [Huge Pages](#)

- [Memory Order Violation](#)
- [Memory Alignment](#)
- Bad Speculation:
  - [Branches To CMOVs](#)
  - [Conditional Store](#)
  - [Replacing Branches With Lookup Tables](#)
  - [C++ Virtual Calls](#)
- CPU Frontend Bound:
- Data-Driven optimizations:
- Misc:
  - [Warmup](#)
  - [LTO](#)
  - [PGO](#)
  - [Optimize IO](#)

## Support the project

---

Performance Ninja is in a very much work-in-progress state. We will be adding new lab assignments and videos! The course is free by default, but we ask you to support us on [Github Sponsors](#), [Patreon](#) or [PayPal](#). Your sponsorship will speed up adding new lab assignments.

Current sponsors:

- Pavel Davydov (@pdavydov108)
- Matias Christensen
- Maya Lekova (@MayaLekova)
- Aaron St. George (@AaronStGeorge)

Thanks to Mansur Mavliutov (@Mansur) for providing an AMD-based machine for running CI jobs.

Lab authors:

- Andrew Evstyukhin (@andrewestyukhin)
- Ivica Bogosavljevic (@ibogosavljevic)
- René Rahn (@rrahn)
- Adam Folwarczny (@adamf88)
- Jakub Beránek (@Kobzol)
- Jakub Gałecki (@kubagalecki)
- Jonathan Hallstrom (@JonathanHallstrom)

- Oleg Makovski (@Olegmak)
- Denis Bakhvalov (@dendibakh)

## Contributing

---

We warmly welcome contributions! See [Contributing.md](#) for the details