



# Optimizing application performance through optimizing compilation

---

Francois Flandin

Encadré par  
Pr Sid Touati

First semester of year 2024-2025

## Objective

Explore how compilers optimizes programs using several optimizations levels : -O0 , -O1 , -O2 , -O3 , -Os .

## 2 parts to the project

1. Compiling 2 programs in C/C++ with each optimization level and compiler to compare performances
2. Checking which optimization is enabled for each optimization level and compiler

**Experience**

Several methods are available on linux

/cpu folder

Navigating the folder can give information about processor's topology, cache levels, number of cpus, cache repartition, etc.

lscpu command

Pas encore fait

other tools

Pas encore fait

**Model name :** 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz

**Adress size :** 39 bits physical, 48 bits virtual

**Cache line size :** 64 bytes

**Cores :** 4

Graphical Topology				
Cores	0 4	1 5	2 6	3 7
L1 Cache	48 kB	48 kB	48 kB	48 kB
L2 Cache	1MB	1MB	1MB	1MB
L3 Cache	8 MB			

**Table** – Computer's topology

## Configuration

Computer in a lightweight configuration, avoid OS's optimizations and bloat from other programs or graphical interface.

## OS and compilers

**OS :** Fedora Linux Workstation v40

**gcc :** version 14.2.1

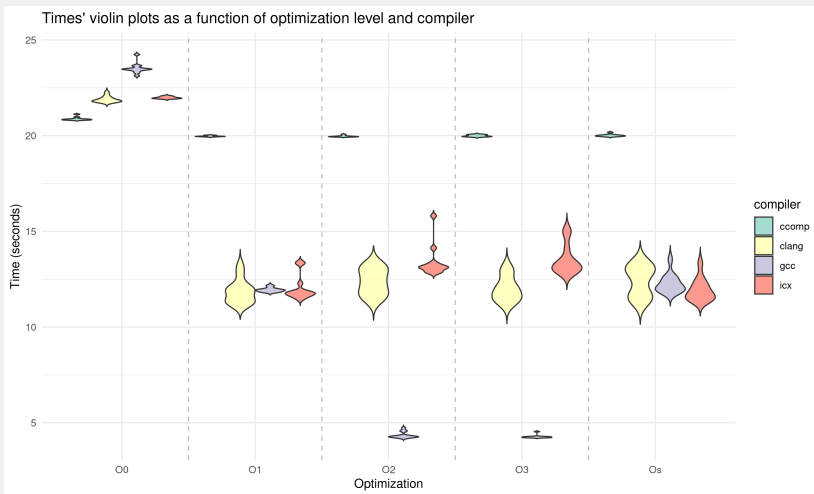
**icx :** version 2024.2.1

**clang :** version 18.1.8

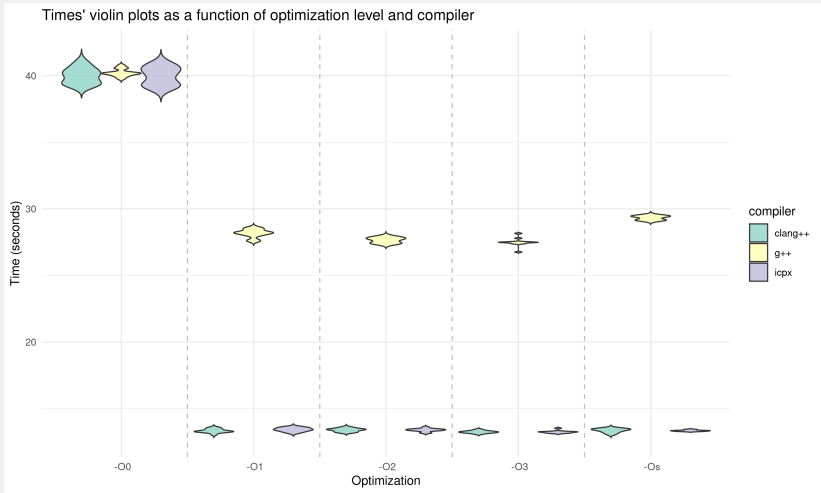
**ccomp :** version 3.14

## Benchmark results





**Figure** – Evolution of the execution time of the program `mat_mult.c` as a function of compiler and optimization level.



**Figure** – Evolution of the execution time of the program `mat_mult.c` as a function of compiler and optimization level.