**Київський національний університет імені Тараса Шевченка**

**факультет радіофізики, електроніки та комп’ютерних систем**

**Комп’ютерні системи**

Лабораторна робота № 3

# Тема: «Дослідження оптимізації коду з використанням векторних розширень CPU»

Роботу виконав

студент 3 курсу

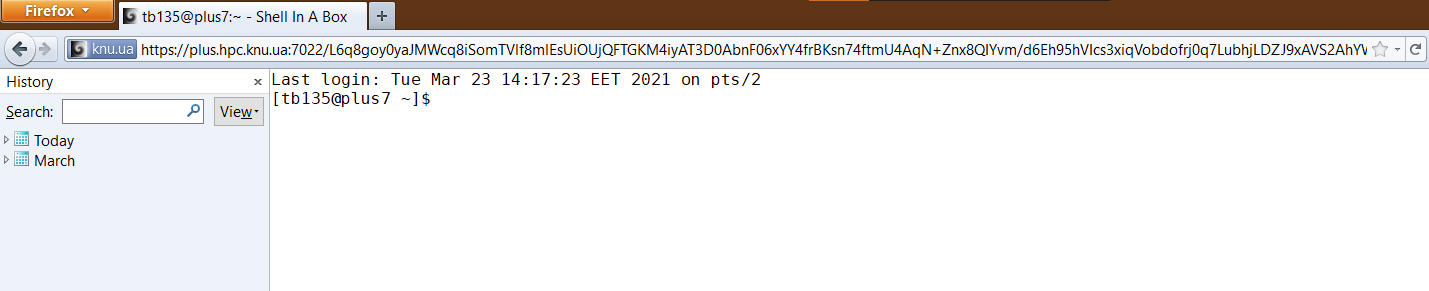
спеціальності “КІ-СА”

Ситниченко Денис Вікторович

Київ 2021

**Хід виконання роботи:**

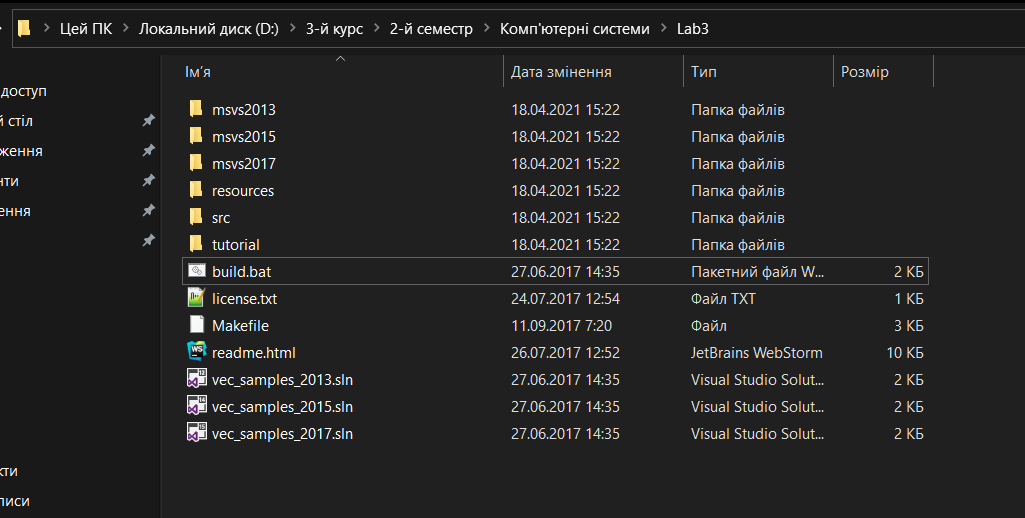
## **Отримайте доступ на обчислювальний кластер для роботи з Intel Compiler**

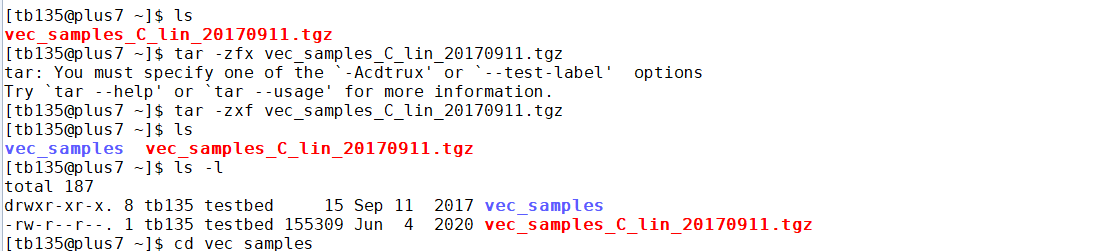
Доступ було отримано

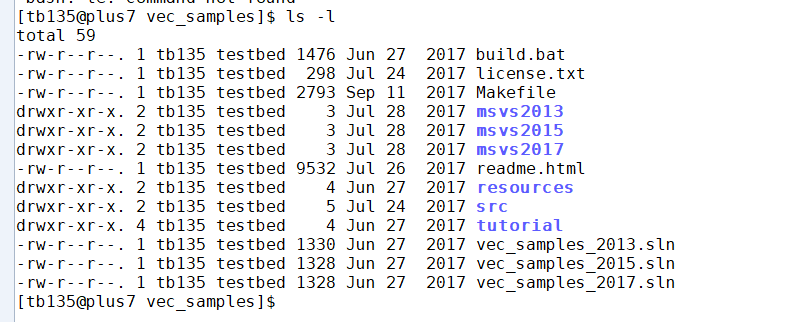
## **Завантажте файли Intel® C++ Compiler - Using Auto-Vectorization Tutorial**

Кластер:

ПК:

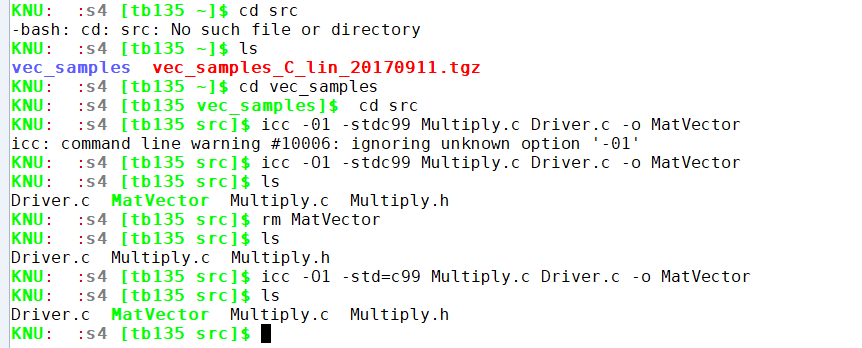


Архів було розархівовано на кластері:

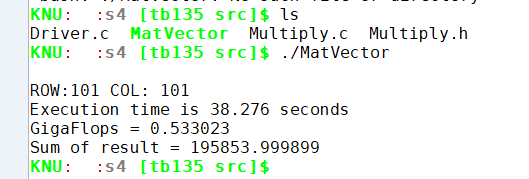


## **Було здійснено вхід в робочу ноду, та завантажено оточення компілятору**

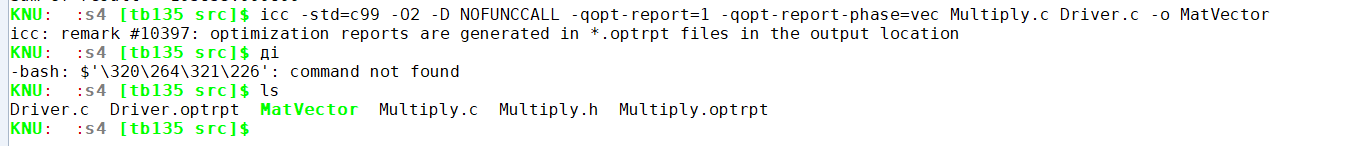
Було виконано компіляцію коду в директорії src без оптимізацій

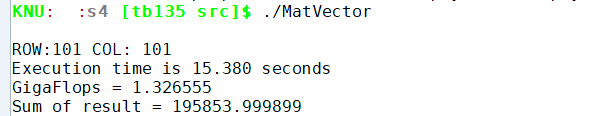


Після чого запущено програму MatVector

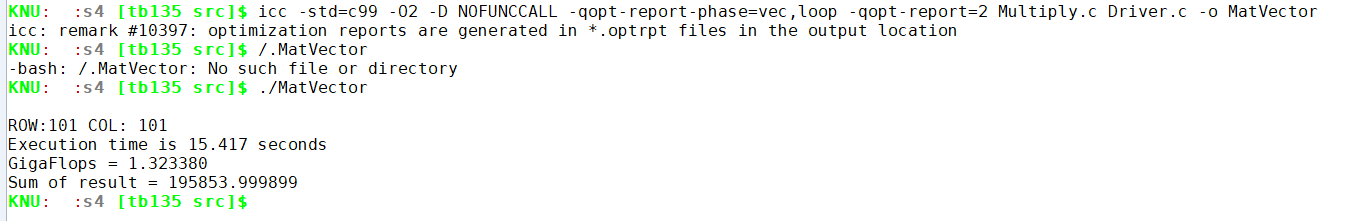


Час виконання без оптимізацій склав: 38.276 секунд.

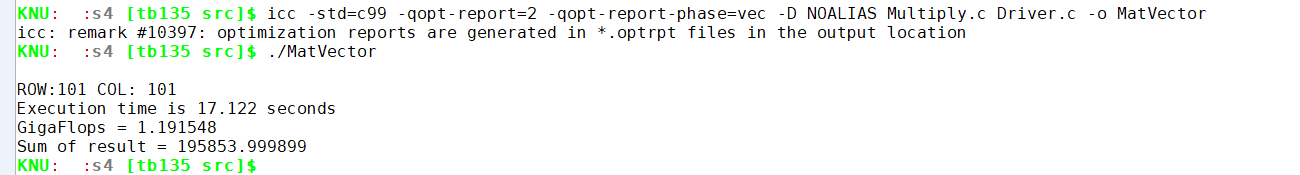
Було виконано компіляцію з параметром O2, та згенеровано звіт по векторизації



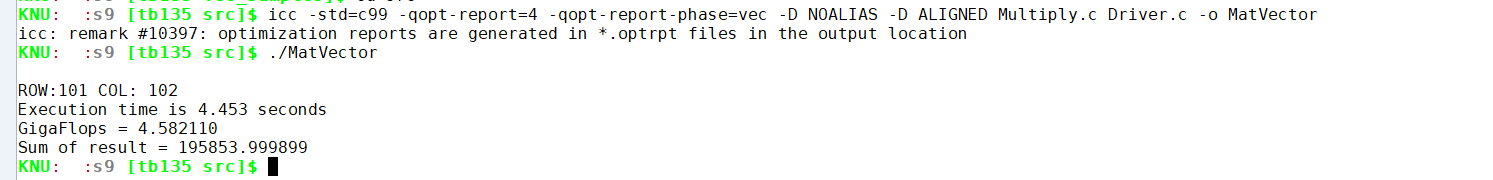
Час виконання з оптимізацією склав 15.380 секунд.

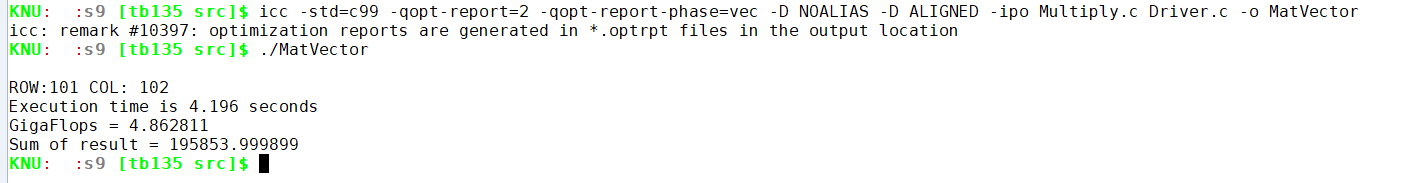
Було виконано компіляцію з параметром O2 та параметрами qopt-report=2 та qopt-report-phase=vec,loop options.

Час виконання склав 15.417 секунд.

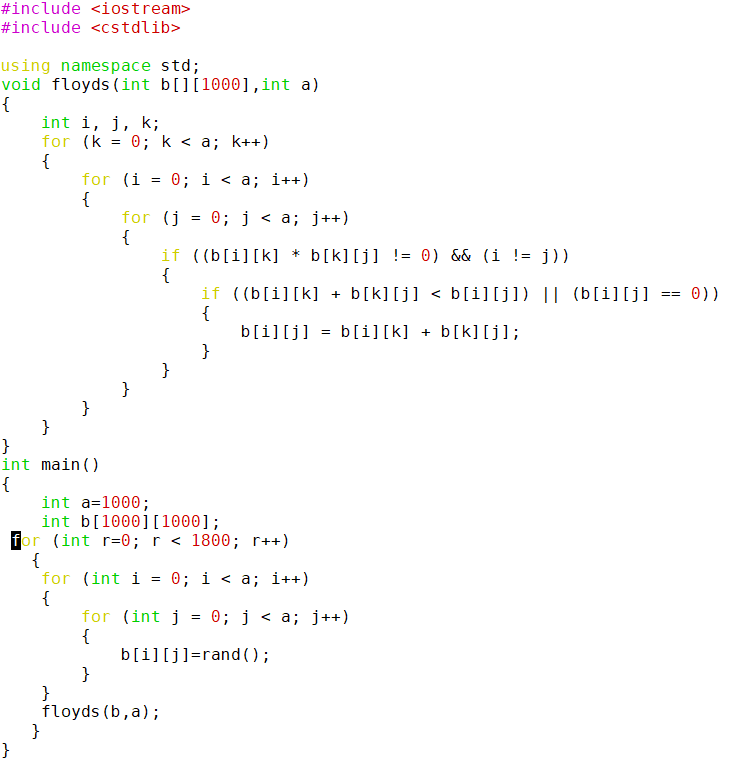
Проект було скомпільовано з вилученням -D NOFUNCCALL щоб відновити виклик matvec(), потім додано -D NOALIAS опцію до командного рядку.

Час виконання з NOALIAS склав 17.122 секунд

Проект було скомпільовано та запущено виконання тестової програми з NOALIAS та ALIGNED оптимізацією.

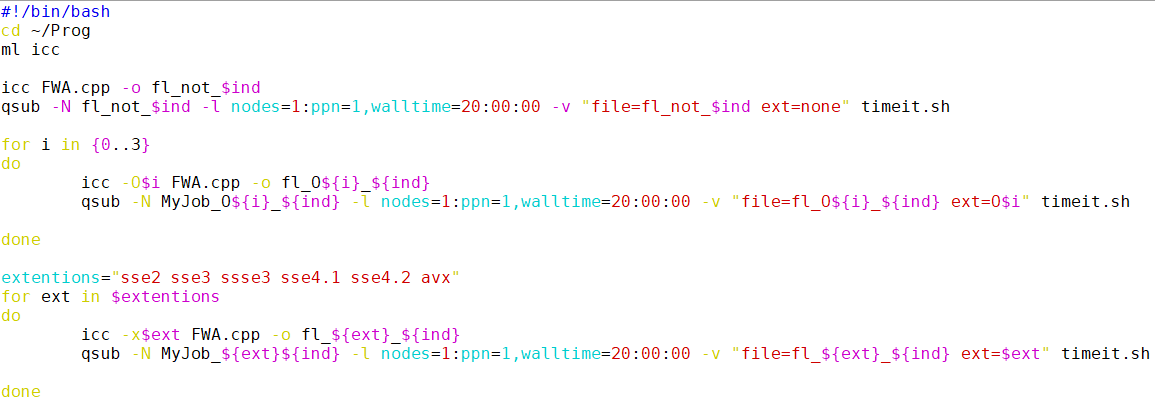
Проект було зкомпільовано з опцією –ipo(міжпроцедурною оптимізацією)

## **Оберіть будь-яку неінтрерактивну консольну програму мовою С/C++**

Було обрано програму мовою с++, яка за алгоритмом Флойда — Воршелла знаходить найкоротший шлях у зваженому графі

Було написано 2 сценарії, перший виконує компіляцію програми з різними оптимізаціями та викликає 2-ий сценарій, який в свою чергу виводить час виконання.

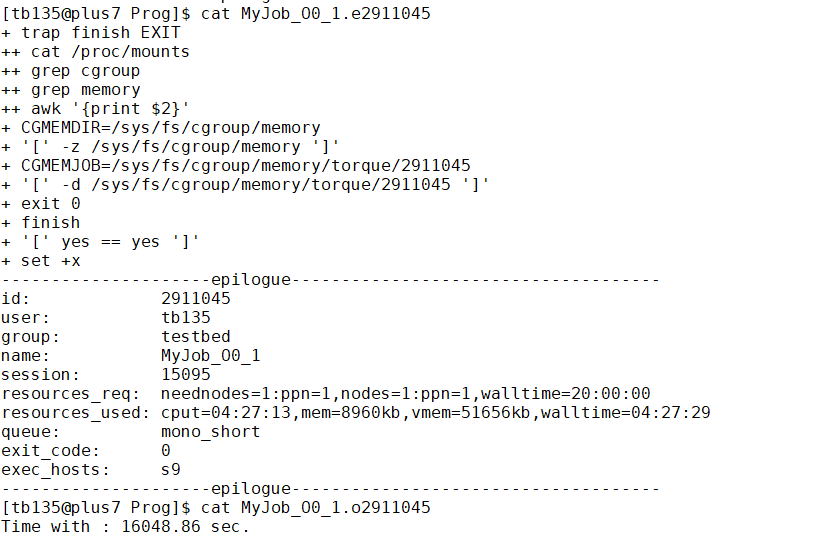
****Скріпт для виводу часу виконання:

Скріпт****, який виконує компіляцію програми з різними оптимізаціями**:**

Скріпти та код, який виконується було завантажено в репозиторій:

<https://gl.vlabs.knu.ua/frecs/ce/cs/2020-2021/DenysS/lab3>

Було запущено сценарій 5 разів, після чого перенесено час виконання в таблицю для порівняння.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ сценарію** | **1** | **2** | **3** | **4** | **5** |
| **Без оптимізації** | 12283,64 | 25948,48 | 25840,10 | 9328,69 | 19047,33 |
| **O0** | 16048,86 | 33699,67 | 33936,13 | 20781,63 | 21136,71 |
| **O1** | 6615,84 | 4074,10 | 4245,78 | 6238,85 | 3729,60 |
| **O2** | 23735,93 | 5338,87 | 8949,11 | 23132,87 | 5176,79 |
| **O3** | 16131,63 | 12380,79 | 23117,67 | 17621,16 | 5667,51 |
| **SSE2** | 17368,61 | 19036,35 | 19027,99 | 19153,23 | 17650,98 |
| **SSE3** | 6036,27 | 19164,88 | 19123,69 | 19064,92 | 4947,10 |
| **SSSE3** | 26537,39 | 19100,14 | 19760,12 | 26027,08 | 19083,96 |
| **SSE4.1** | 26529,53 | 19651,57 | 19799,95 | 26023,73 | 19074,71 |
| **SSE4.2** | 25955,27 | 19659,25 | 19668,72 | 26128,60 | 24711,81 |
| **AVX** | 25846,07 | 19656,28 | 19161,34 | 26131,31 | 24716,41 |

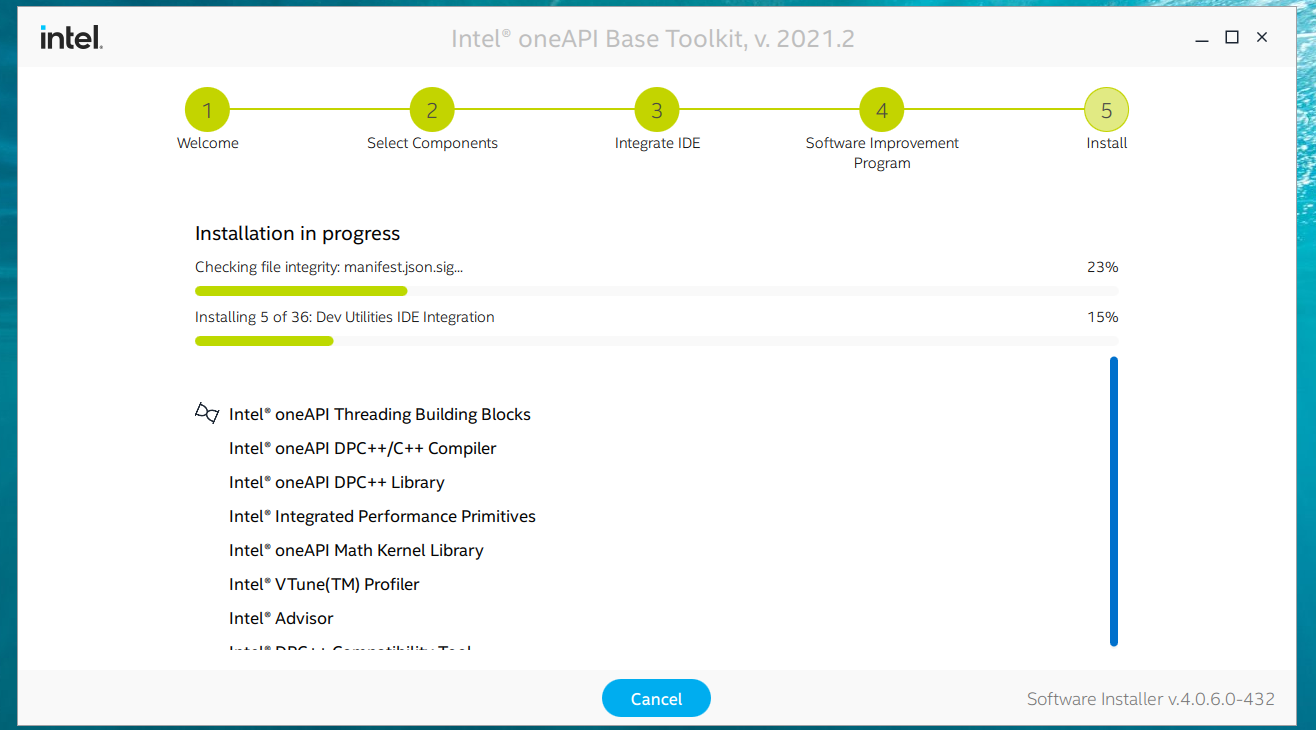
Час виконання відповідно до застосованої оптимізації в секундах:

Після було обраховано середнє значення (також в секундах) та стандартне відхилення з урахуванням викидів(найбільшого та найменшого значення в оптимізації). Викиди були видалені, що можна побачити в таблиці:

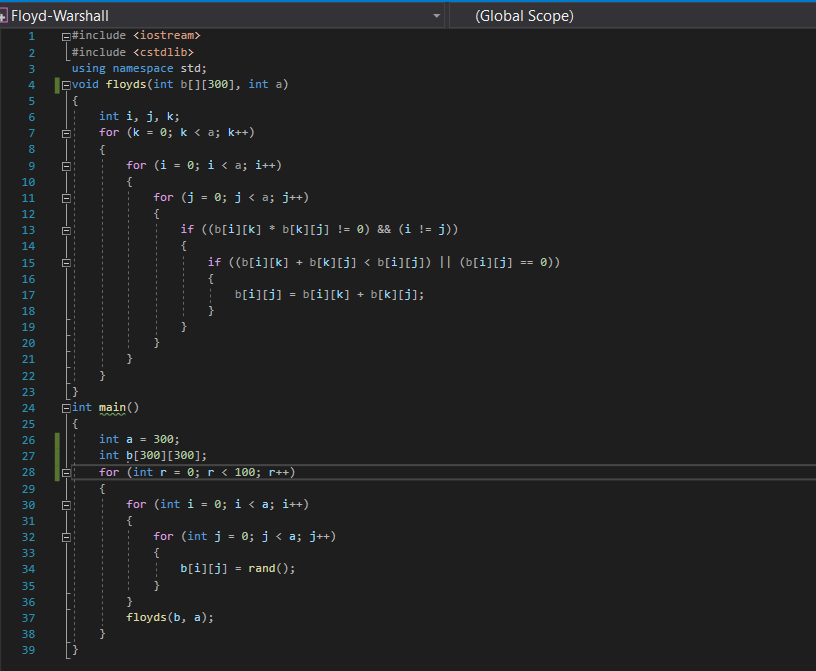
Графічне представлення даних:

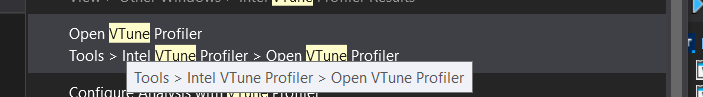
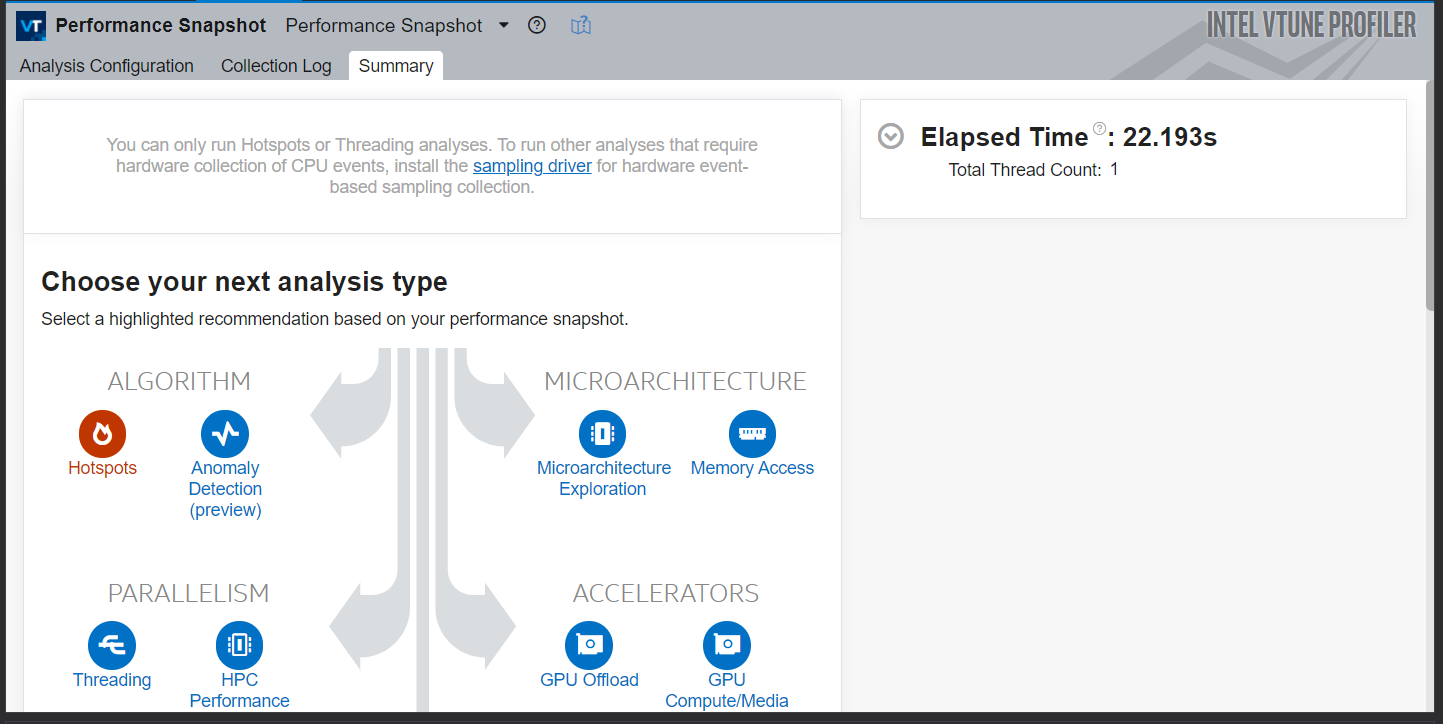
Згідно графіків найшвидший час виконання при оптимізації О1(в середньому 4852,91 секунди) , а найдовший при О0 (в середньому 25206,00 секунди)

## **Оптимізація з використанням Intel® Parallel Studio**

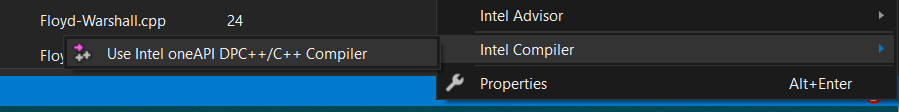
****Було встановлено Intel oneAPI Base Toolkit

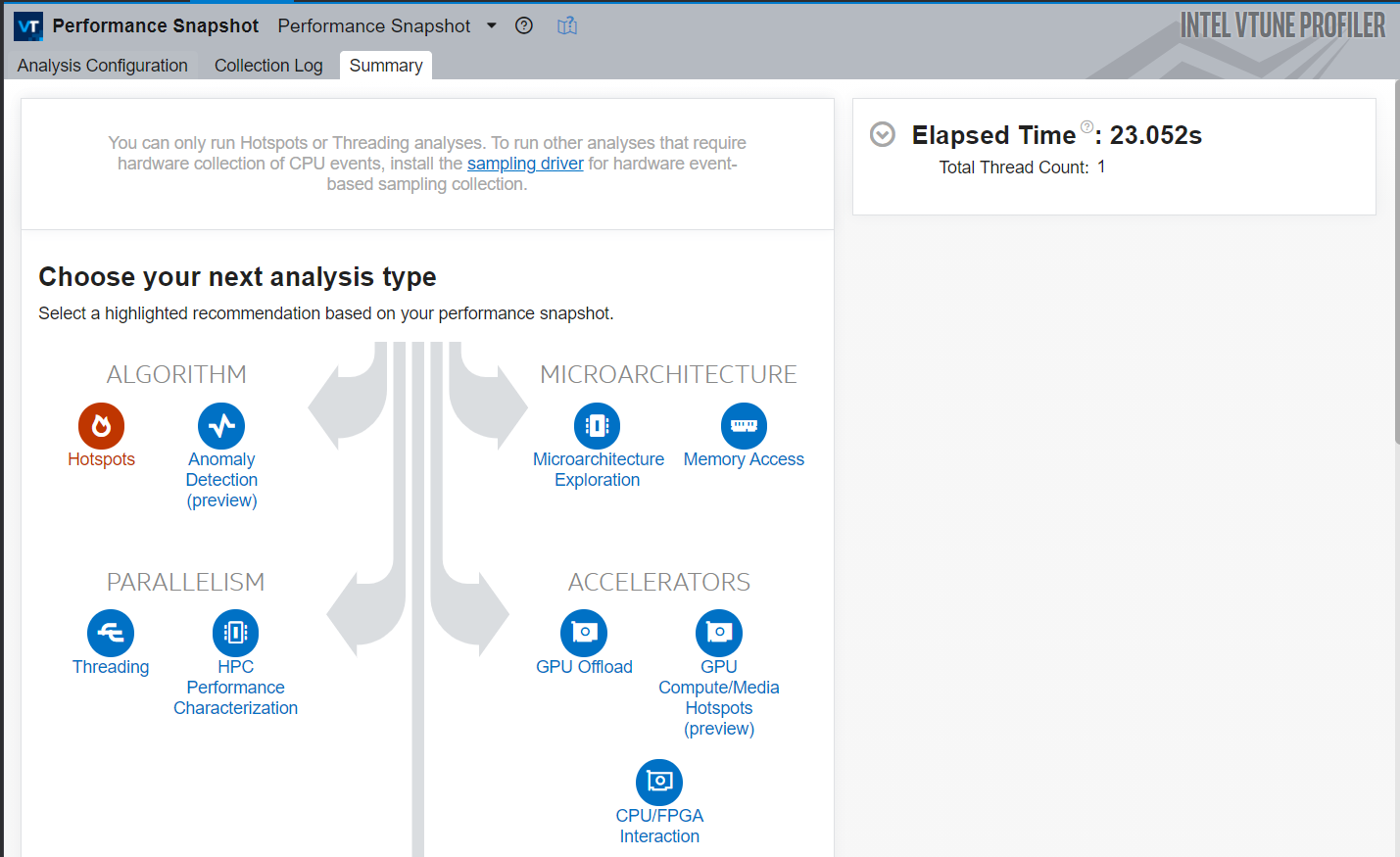
Програму було модифіковано для оптимізації на робочій машині:



**** Після чого було скомпільовано без оптимізацій, та проведено аналіз за допомогою vtune:

Час роботи склав 22.193 секунди.

****Після чого той самий код було скомпільовано через Intel Compiler

****Як видно компілятор він Intel витратив більше часу, а саме: 23.052 секунди

**Висновок**: різні види оптимізації по-різному впливають на швидкодію програми, можуть як і пришвидшити виконання так і уповільнити в рази, що було продемонстровано в порівняльній таблиці. Також використання різних компіляторів може дати як і прибавку до швидкодії так і уповільнити її.