**Отчет по лабораторной работе №4**

*Островского Артема ИВТ-11М*

**Задание 1.**

Необходимо собрать проект с исходным файлом graph.cpp и запустить собранный исполняемый файл. Оценить время работы программы и корректность ее работы. Зафиксируйте это в отчете. Оцените сложность алгоритма.

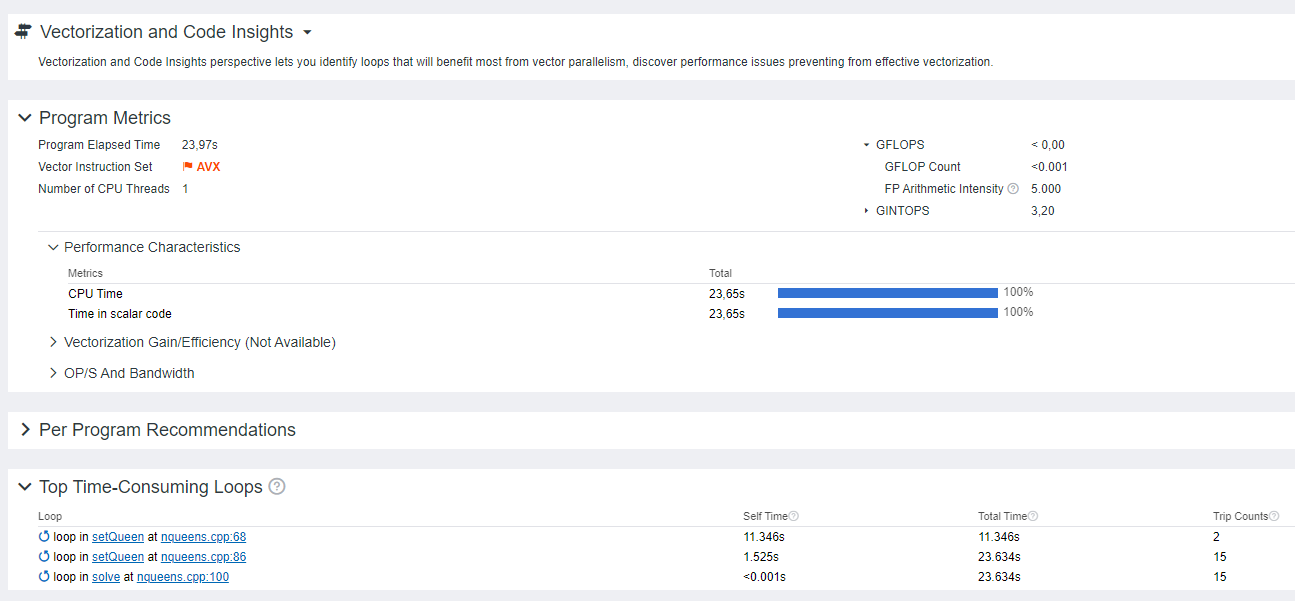
*#запуск программы без оптимизации*

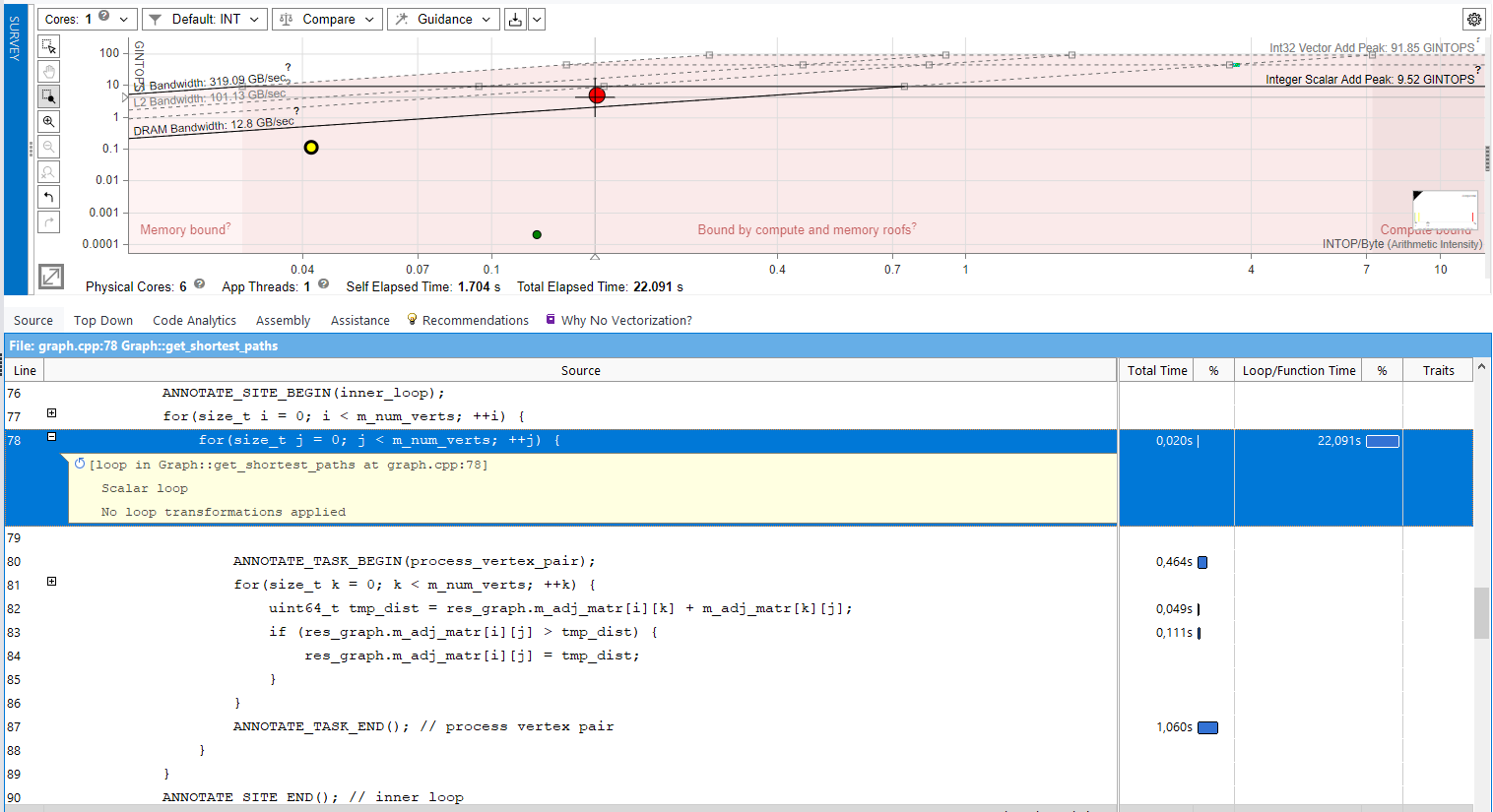
|  |  |  |
| --- | --- | --- |
| Usage: graph.exe vertex\_num [default is 400].  - 5 8  3 - 1  6 2 -  Test run started  Test results:  - 5 6  3 - 1  5 2 -  Perf run started  Elapsed perf time: **20.0487** | Usage: graph.exe vertex\_num [default is 400].  - 5 8  3 - 1  6 2 -  Test run started  Test results:  - 5 6  3 - 1  5 2 -  Perf run started  Elapsed perf time: **20.1137** | Usage: graph.exe vertex\_num [default is 400].  - 5 8  3 - 1  6 2 -  Test run started  Test results:  - 5 6  3 - 1  5 2 -  Perf run started  Elapsed perf time: **19.8066** |

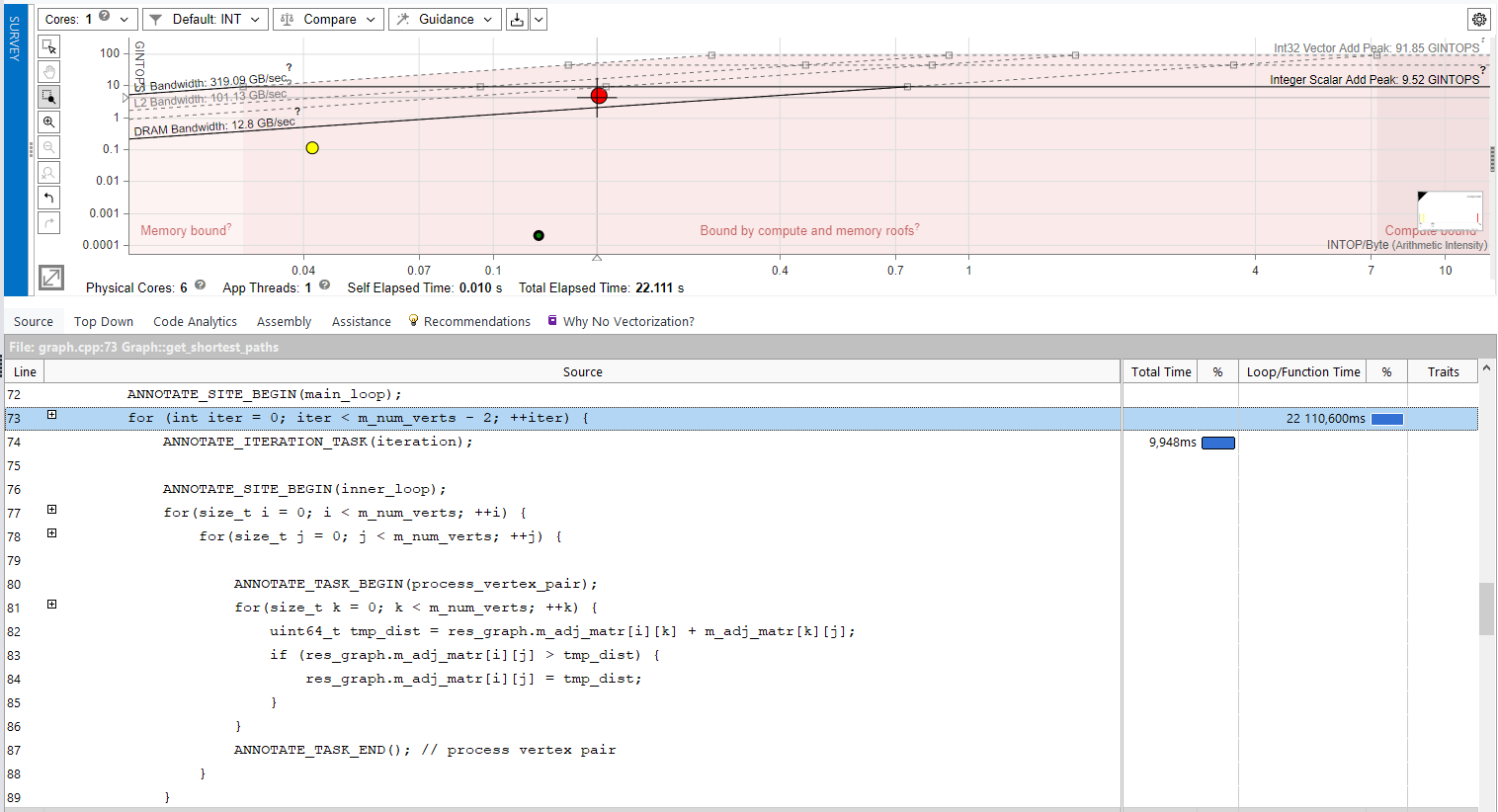
Сложность алгоритма: , так как алгоритм сравнивает все возможные пути через граф между каждой парой вершин.

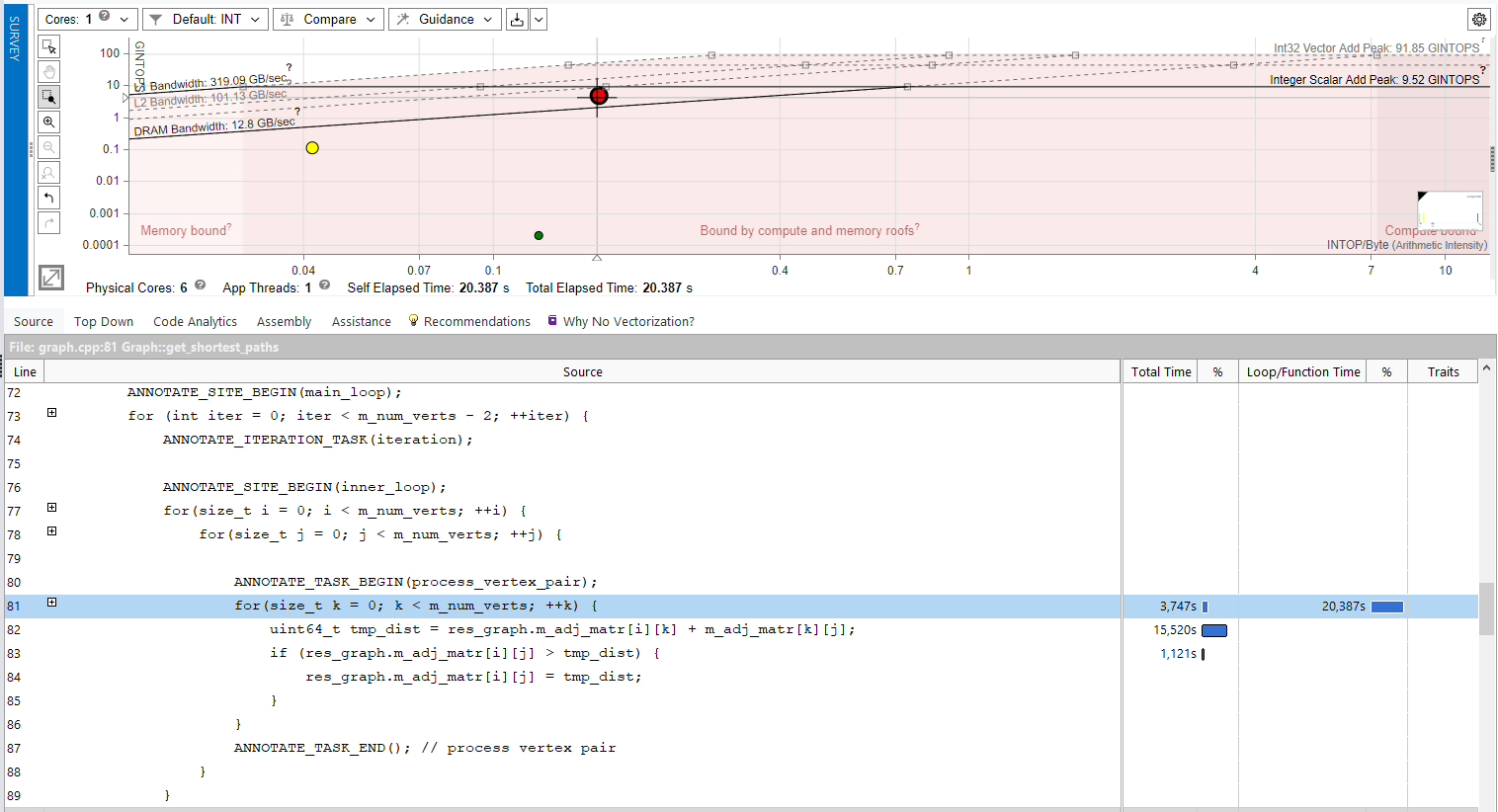
**Задание 2.**

С помощью инструментария Advisor необходимо получить метрики исполнения программы в последовательном режиме: GFLOPs, количество используемых потоков, тип векторных команд, построить roofline, зафиксировать список хотспотов, выделить ключевые метрики, сохранить snapshot.



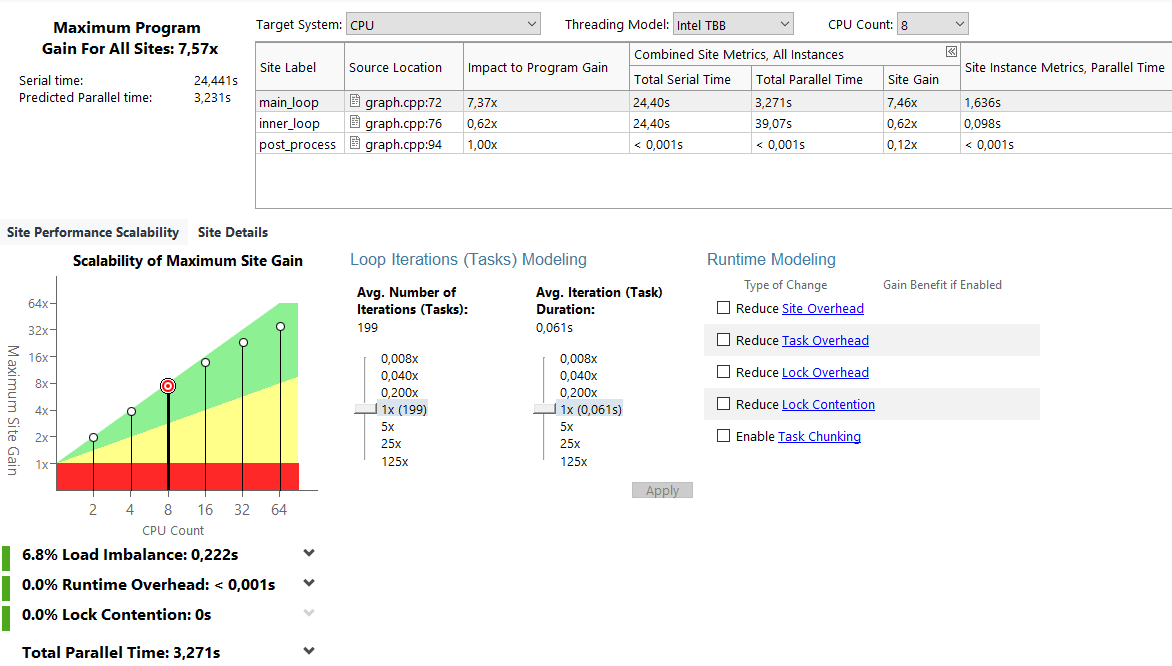
Hotspot #1

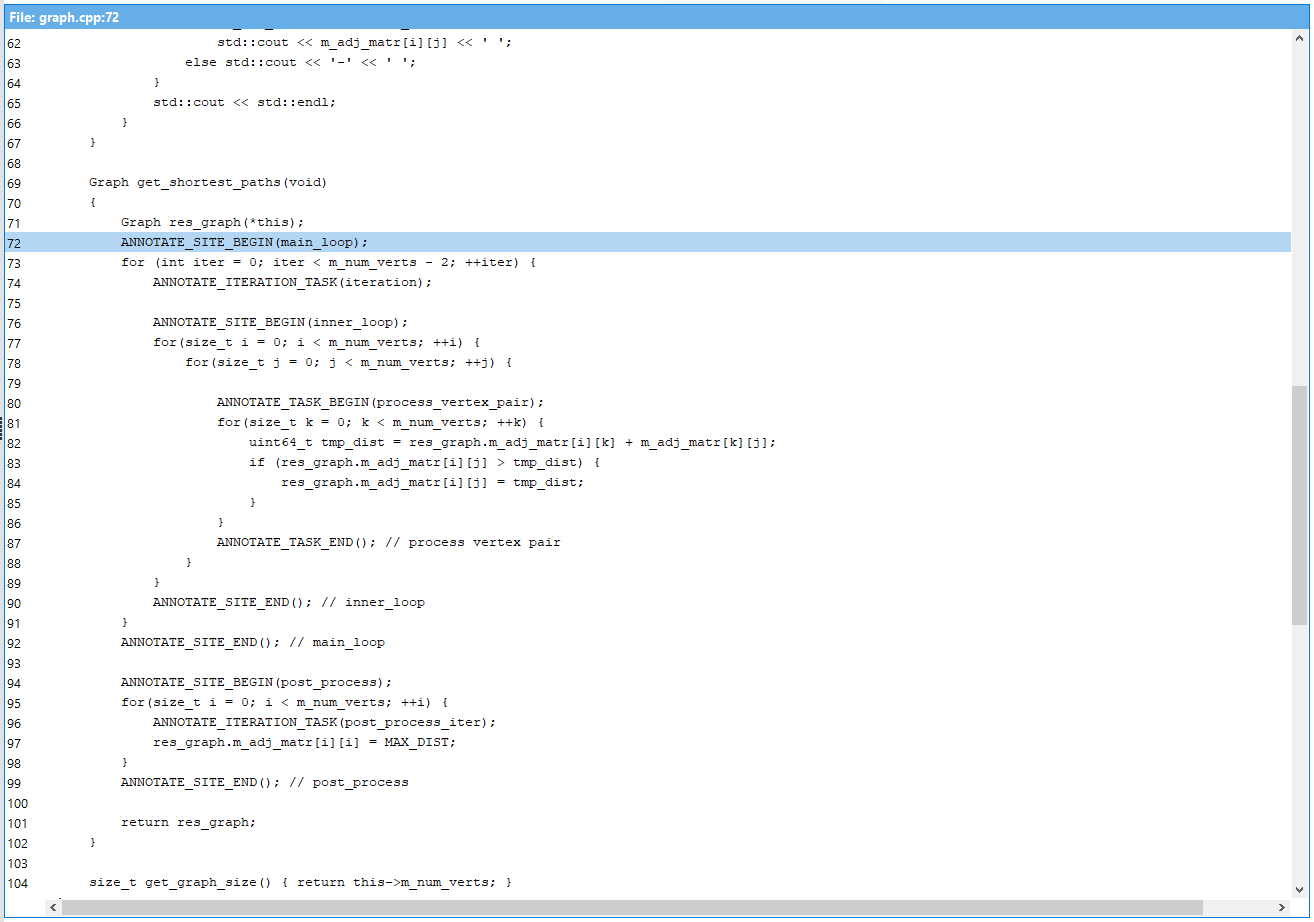
Hotspot #2

Hotspot #3

**Задание 3.**

Максимально ускорить выполнение программы, убедиться в корректности ее работы, построить совместные roofline для разных решений, оценить, во сколько произошло ускорение. Для сдачи лабораторной работы на полный балл необходимо получить ускорение для последовательной версии программы (без добавления параллелизма, за счет оптимизаций), а затем и с добавлением параллелизма. Зафиксировать оба шага в отчете.

Выполнение suitability-анализа



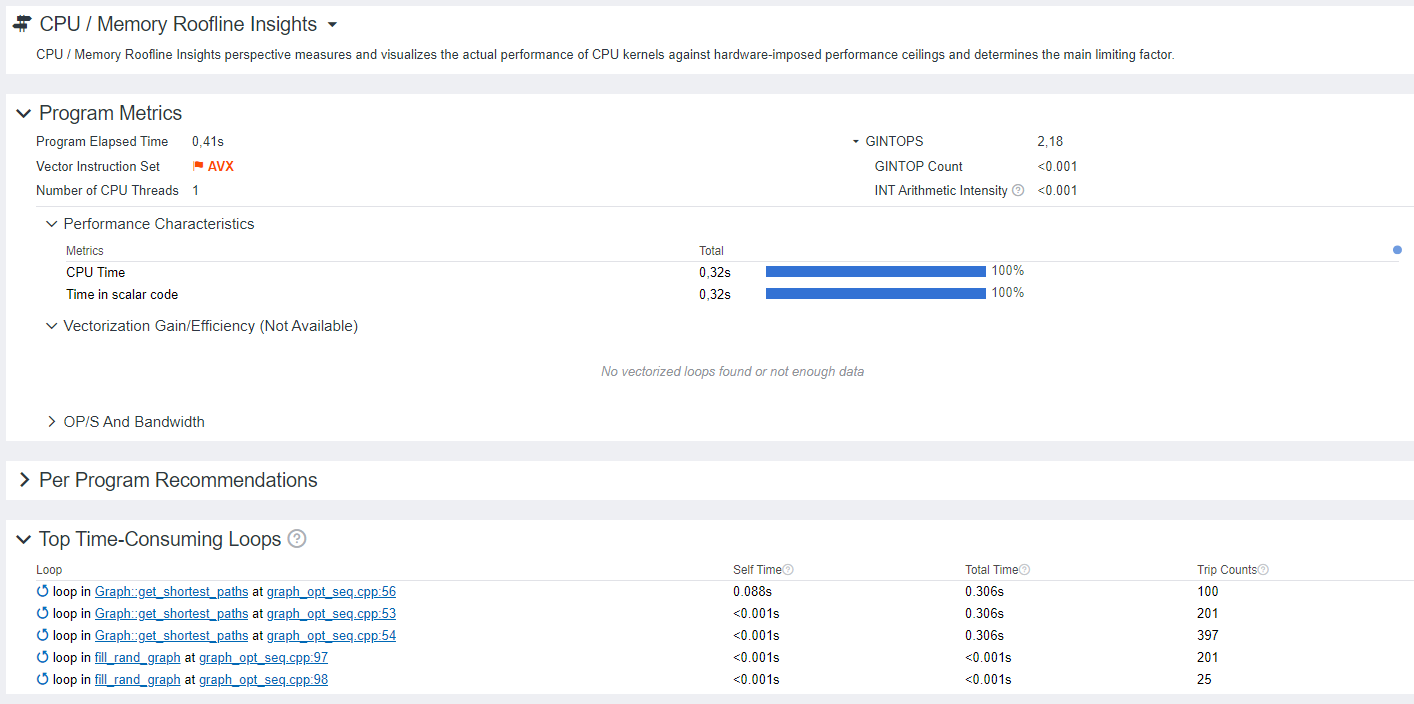
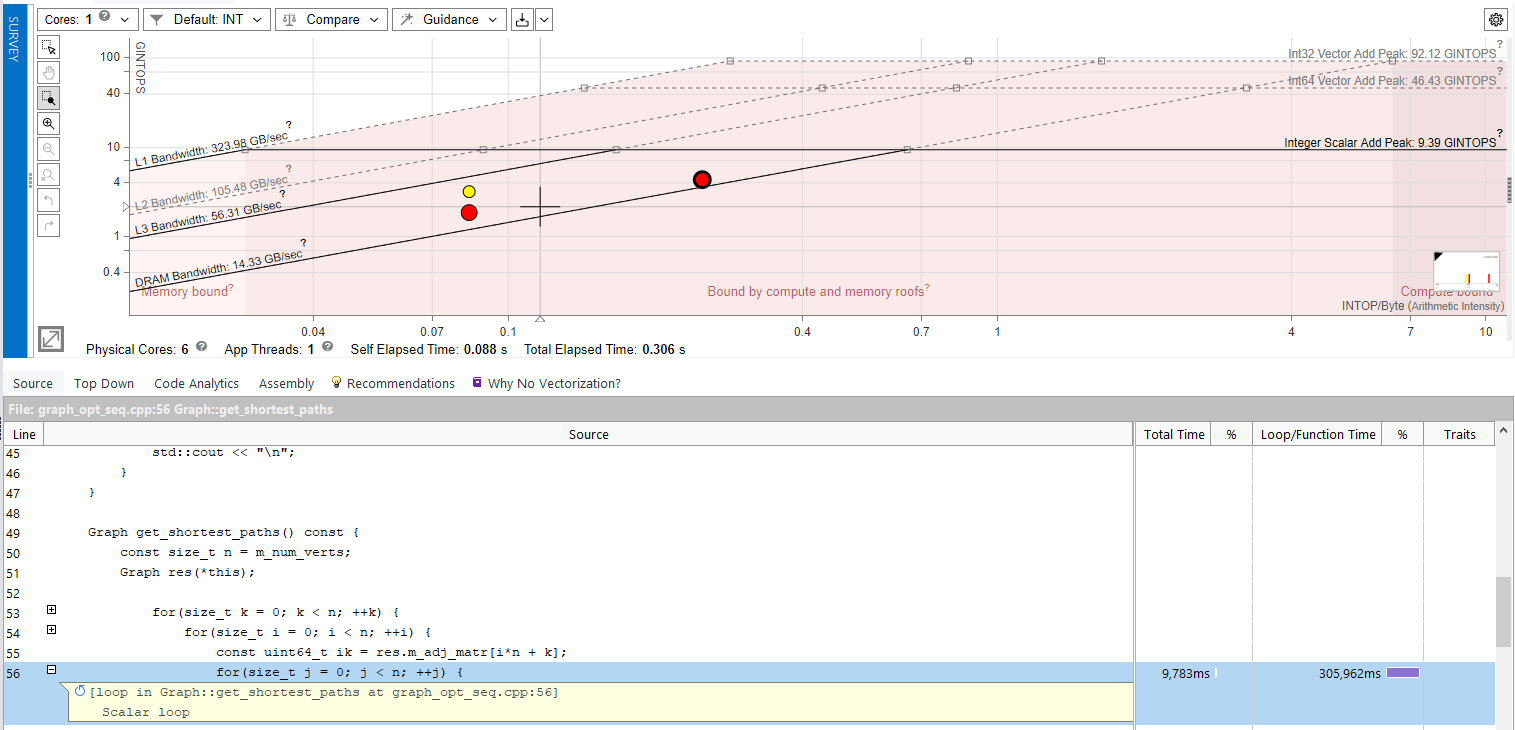
Оптимизация последовательного алгоритма:

- раскручивание двумерного массива в одномерный вектор;

- изменение порядка циклов.

Результаты оптимизации:

|  |  |  |
| --- | --- | --- |
| Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  3 0 1  5 2 0  Perf run started  Elapsed perf time: **0.303982** | Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  3 0 1  5 2 0  Perf run started  Elapsed perf time: **0.299942** | Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  3 0 1  5 2 0  Perf run started  Elapsed perf time: **0.303215** |



Добавление параллелизма

|  |  |  |
| --- | --- | --- |
| Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  3 0 1  5 2 0  Perf run started  Elapsed perf time: **0.0820981** | Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  3 0 1  5 2 0  Perf run started  Elapsed perf time: **0.0818417** | Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  3 0 1  5 2 0  Perf run started  Elapsed perf time: **0.0794323** |

|  |  |  |
| --- | --- | --- |
| Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  5 2 0  Perf run started  Elapsed perf time: 0.084616 | Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  3 0 1  5 2 0  Perf run started  Elapsed perf time: **0.0747733** | Usage: graph.exe vertex\_num [default is 400].  0 5 8  3 0 1  6 2 0  Test run started  Test results:  0 5 6  3 0 1  5 2 0  Perf run started  Elapsed perf time: **0.076357** |

