Отчет по лабораторной работе №2

Сайфуллин Джамиль ИВТ-11М

Технические характеристики

OS: Windows 10

Vs: Visual Studio 2017 v15.9.18

Ips: v2019

Железо: 2 ядра физических, 4 логических 3.4ГГц

# 

# Задание 1-2

В файле task\_for\_lecture3.cpp приведен код, реализующий последовательную версию метода Гаусса для решения СЛАУ. Результат работы программ представлен на рисунке 1.

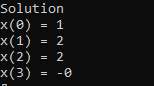


Рисунок 1 Метод Гаусса для решения СЛАУ

Проверим правильность работы программы. Результат на рисунке 2

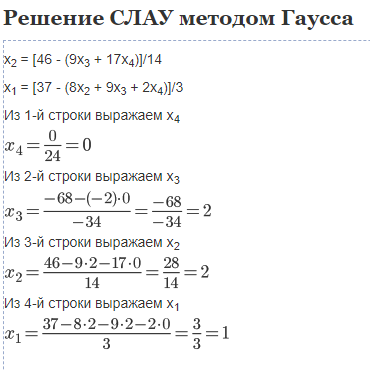


Рисунок 2 Решение СЛАУ методом Гаусса

Результаты совпали.

Теперь запустим программу с размером матрицы MATRIX\_SIZE и инициализируем ее случайными значениями через функцию InitMatrix. Так же измерим время работы прямого хода. Результат представлен на рисунке 3.



Рисунок 3 Результат работы программы с размером матрицы MATRIX\_SIZE

# Задание 3

Запустим Amplifier, чтобы найти места в коде программы, которые можно распараллелить. Результат на рисунке 4

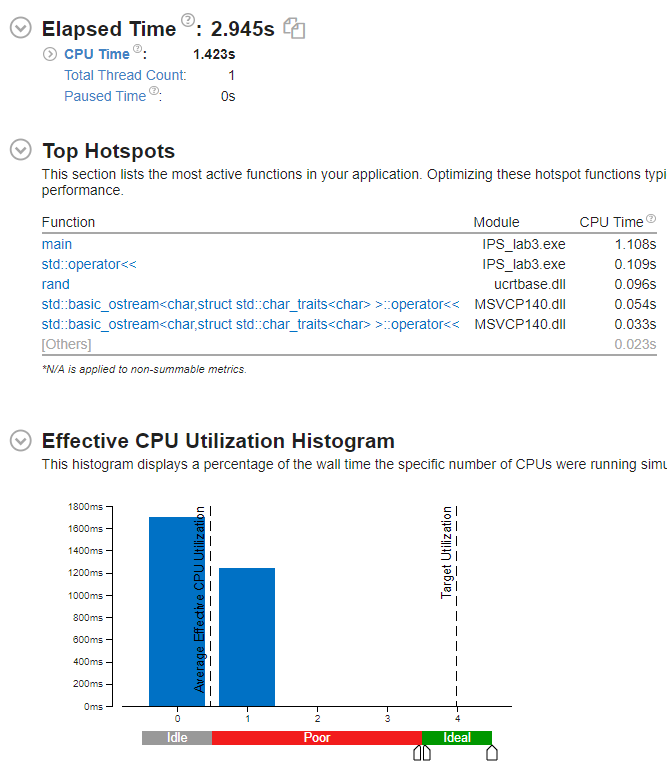


Рисунок 4 Результат работы Amplifier

Создадим на основе последовательной функции SerialGaussMethod новую с использованием cilk\_for, чтобы добиться параллелизма. Результат работы новой функции на рисунке 5.



Рисунок 5. Результат работы ParalleGaussMethod

Результат оказался в разы хуже, чем без параллелизма, вероятно это связано с ошибками гонки данных.

# Задание 4

Воспользуемся ***Inspector XE,*** чтобы найти ошибки. Результат на рисунке 6.

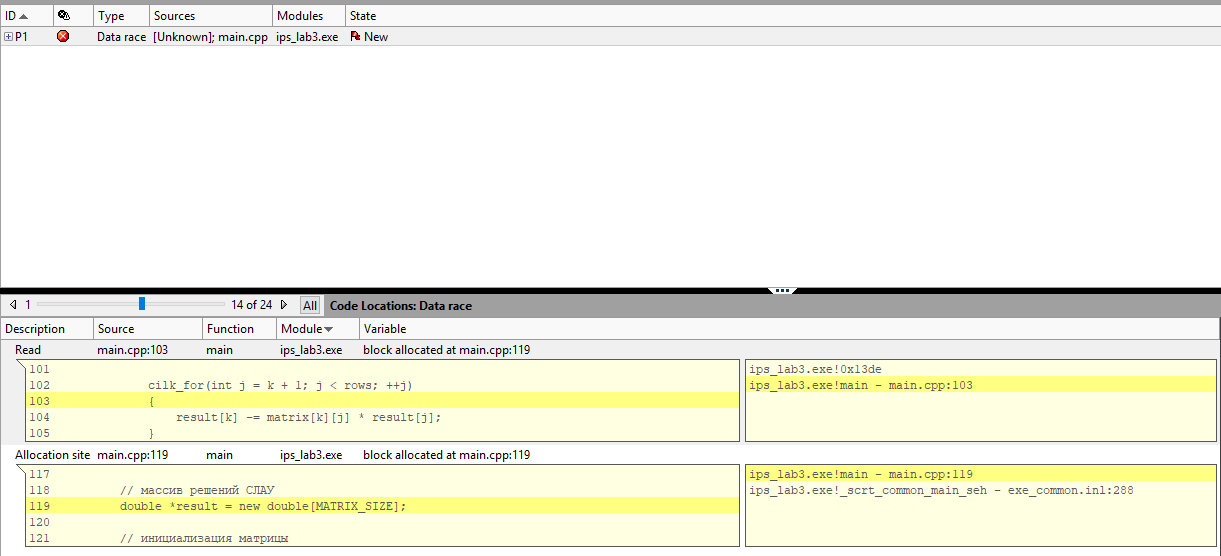


Рисунок 6 результат работы *Inspector XE*

Для исправления ошибки заменим result на reducer\_opadd и устраним гонку данных. Результат исправленной программы на рисунке 7



Рисунок

Воспользуемся ***Inspector XE,*** чтобы найти убедиться в отсутствии гонок данных. Результат на рисунке 8. Как можно видеть, проблема с гонкой данных исчезла. Так же на рисунке 9 и 10 представлены результаты распараллеливания полученный с помощью **Amplifier XE**.

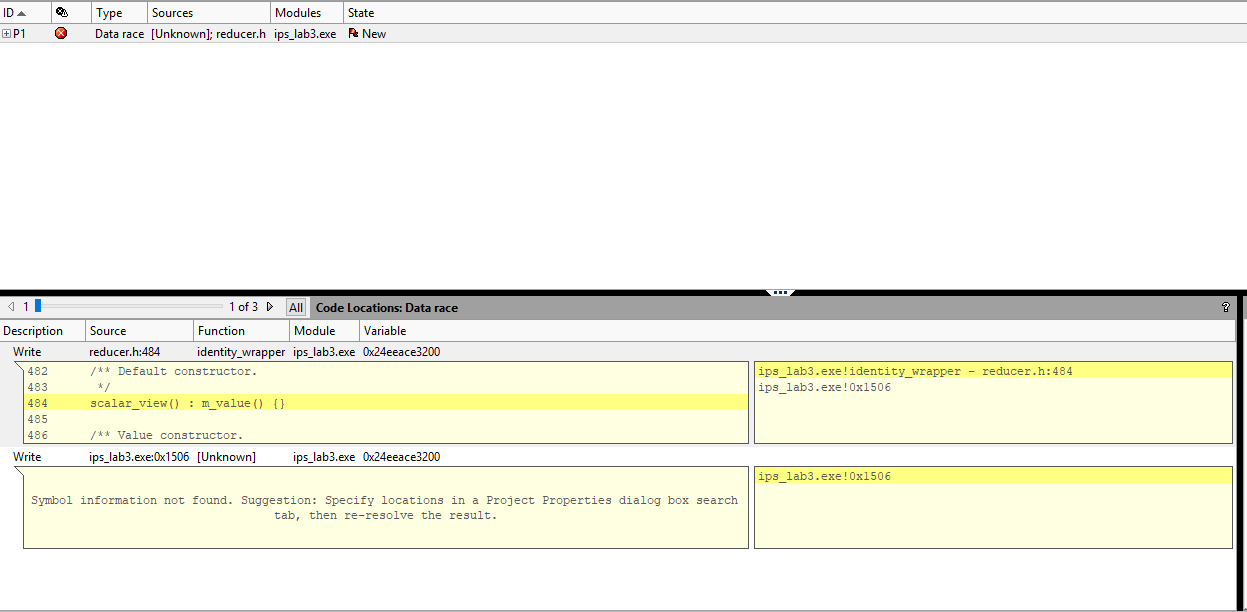


Рисунок 8 Результат работы *Inspector XE*

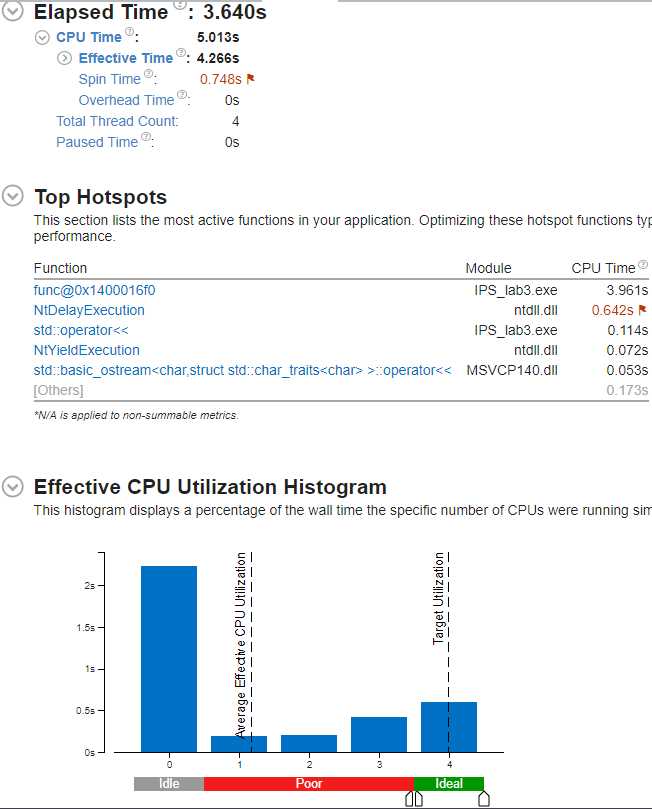


Рисунок 9 Результат работы Amplifier XE



Рисунок 10 Результат работы Amplifier XE

# Задание 5

Удостоверимся, что параллельная функция работает корректно. Используем тестовую матрицу и проверим обе функции. Результат на рисунке 11.

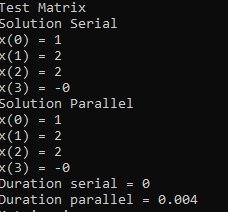


Рисунок 11 Результат работы при тестовой матрице

Функция вычисляет правильно, однако проигрывает по скорости. Это обусловлено тем, что при малом количестве вычислений распараллеливание не дает выигрыша по времени. Запустим функции с матрицей размера MATRIX\_SIZE. Результат на рисунке 12.

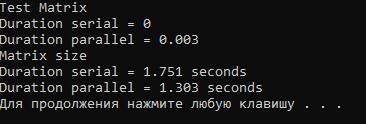


Рисунок 12 Результат работы при тестовой матрице и матрице размером MATRIX\_SIZE

Как можно заметить, при большем объёмы вычислений функция с параллелизмом оказалась быстрее.