

# Разработка параллельных программ с использованием Cilk Plus

Выполнила: Шахманова Мария, ПМ-21м

Напишите параллельную программу вычисления следующего интеграла с использованием дополнений **Intel Cilk Plus** языка C++:

$$\int_{-1}^1 \frac{8}{2+2x^2} dx$$

Код программы необходимо загрузить на **GitHub**. По результатам работы должен быть написан отчет, отражающий методику разработки параллельных программ в среде **Intel Paralle Studio XE** с включенными в него скриншотами экрана.

Отчет должен содержать следующие разделы:

1. Описание проблемы и краткая характеристика инструментов параллелизации, используемых для решения задачи
2. Описание и анализ программной реализации

- Анализ работы программы с использованием **Intel Parallel Inspector XE**;
- Оценка эффективности программной реализации;
- Проверка выполнения работы программы с использованием **Intel VTune Amplifier XE**;
- Сведения о зависимости времени выполнения от заданных параметров алгоритма.

1. Рассмотрим интеграл  $\int_{-1}^1 \frac{8}{2+2x^2} dx$ .

Используем для решения численное интегрирование. Идея численного интегрирования предельно проста и вытекает из геометрического смысла определенного интеграла – значение определенного интеграла численно равно *площади криволинейной трапеции*, ограниченной графиком функции  $y = f(x)$ , осью абсцисс и прямыми  $x=a$ ,  $x=b$ . Отрезок  $[a, b]$  разбивается на  $N$  частичных отрезков, на каждом из которых находится значение функции в точках разбиения и вычисляется площадь криволинейной трапеции. Сумма таких площадей приближённо равна значению интеграла.

Далее будем использовать этот метод

Так как вычисления площадей криволинейных трапеций можно проводить не зависимо друг от друга, используем параллелизм в коде программы. Воспользуемся расширением языка C++ **Intel Cilk Plus**.

В качестве инструментов параллелизации используем следующие:

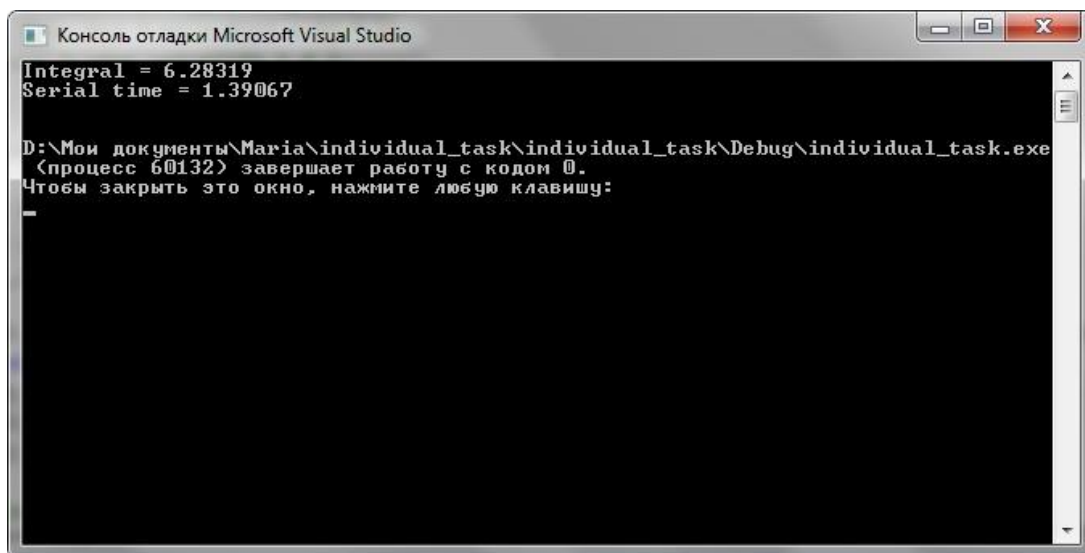
- **Intel Parallel Inspector XE** - анализатор корректности с возможностью проверки работы с памятью и потоками. Проверка памяти включает в себя проверку утечки памяти, повисшие указатели, переменные без инициализации, использование некорректных ссылок на участки памяти, и др. Проверки потоков включают в себя, например, проверки состояний гонки, взаимных блокировок, и др. Таким образом, выявляются основные ошибки, которые могут возникать при разработке параллельного кода.
- **Intel VTune Amplifier** - анализатор производительности, инструмент для сбора и анализа данных о производительности кода (профилировки) последовательных и параллельных приложений (программ) с целью выявления наиболее часто используемых и уязвимых участков кода. Также для параллельных кодов показывает загрузку одновременно работающих процессоров (CPUs).

2. Точное значение интеграла:  $2\pi=6.2832$

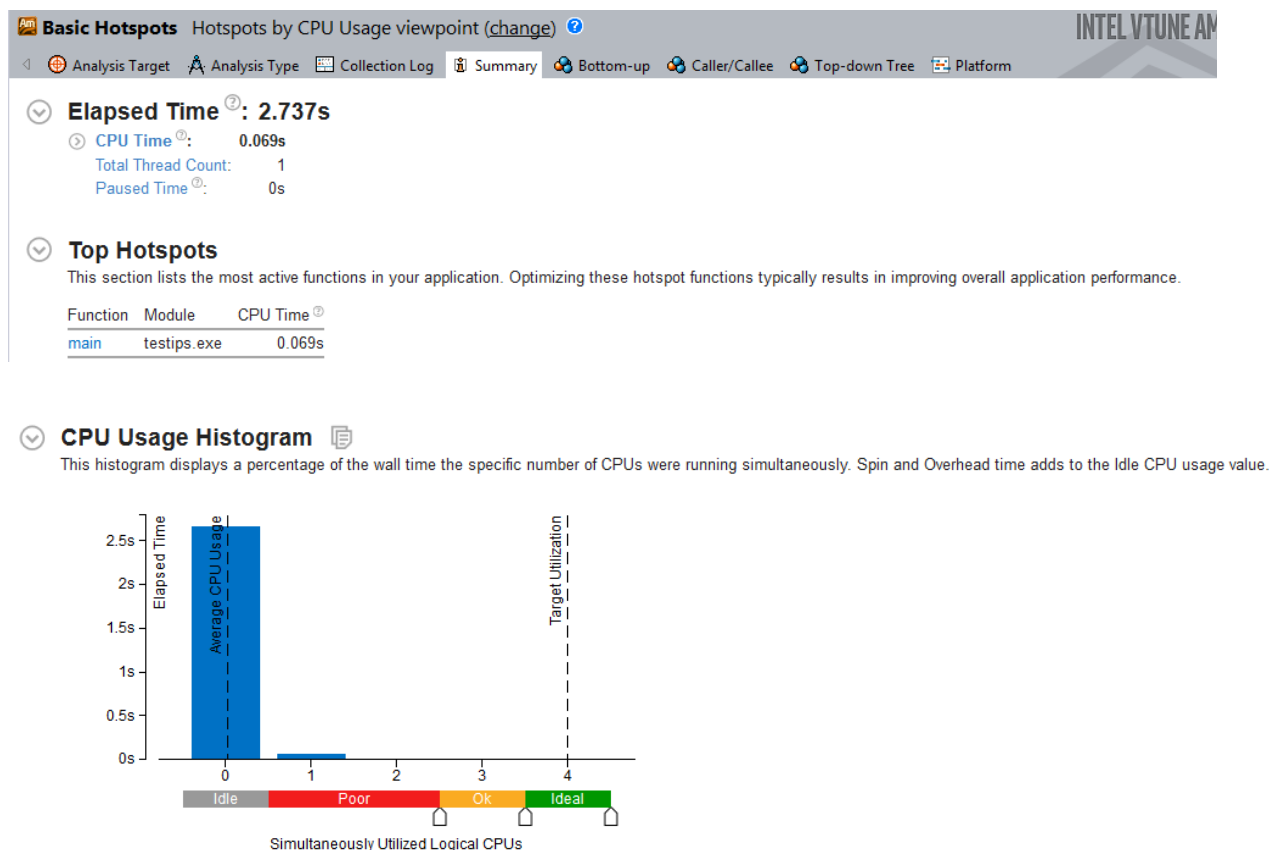
Вспользуемся формулой трапеций (площадь = полусумма оснований \* высоту):

```
{
    double S, h;
    S = 0;
    h = (b - a) / n;
    for (int i = 0; i < n; i++)
        S += (f(a + i * h) + f(a + (i+1) * h));
    return 0.5*S*h;
}
```

Запустим программу и убедимся в корректности вычисления:



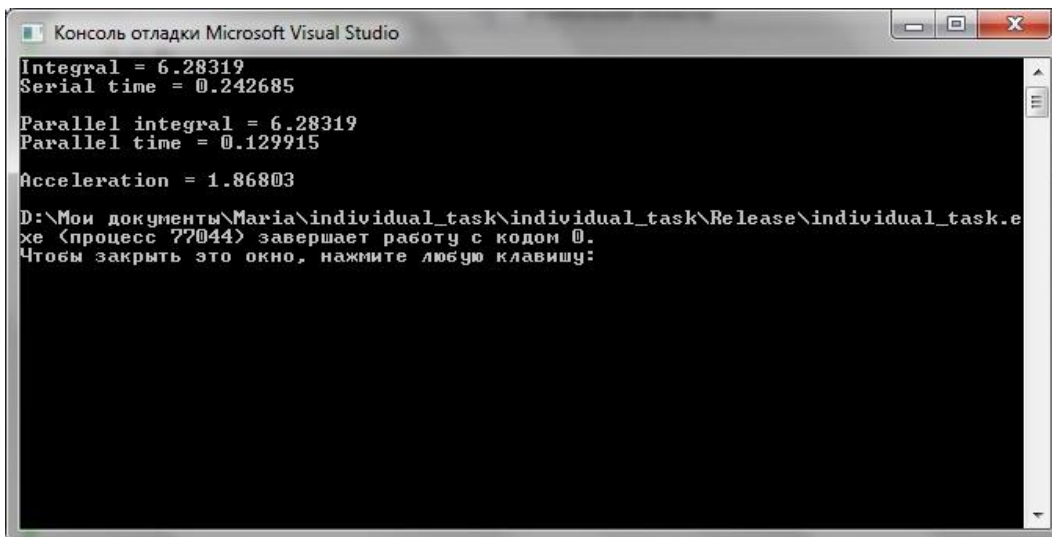
Проанализируем производительность и наиболее повторяющиеся участки кода с помощью *Intel VTune Amplifier*:



Реализуем метод трапеций параллельно с помощью `cilk_for`:

```
double parallel_trapez(int n)
{
    double h = (b - a) / n;
    cilk::reducer_opadd<double> S(0.0);
    cilk_for(int i = 0; i < n; i++)
        S += f(a + i * h) + f(a + (i + 1) * h);
    return S->get_value()*h * 0.5;
}
```

Запустим программу:



```
Консоль отладки Microsoft Visual Studio
Integral = 6.28319
Serial time = 0.242685

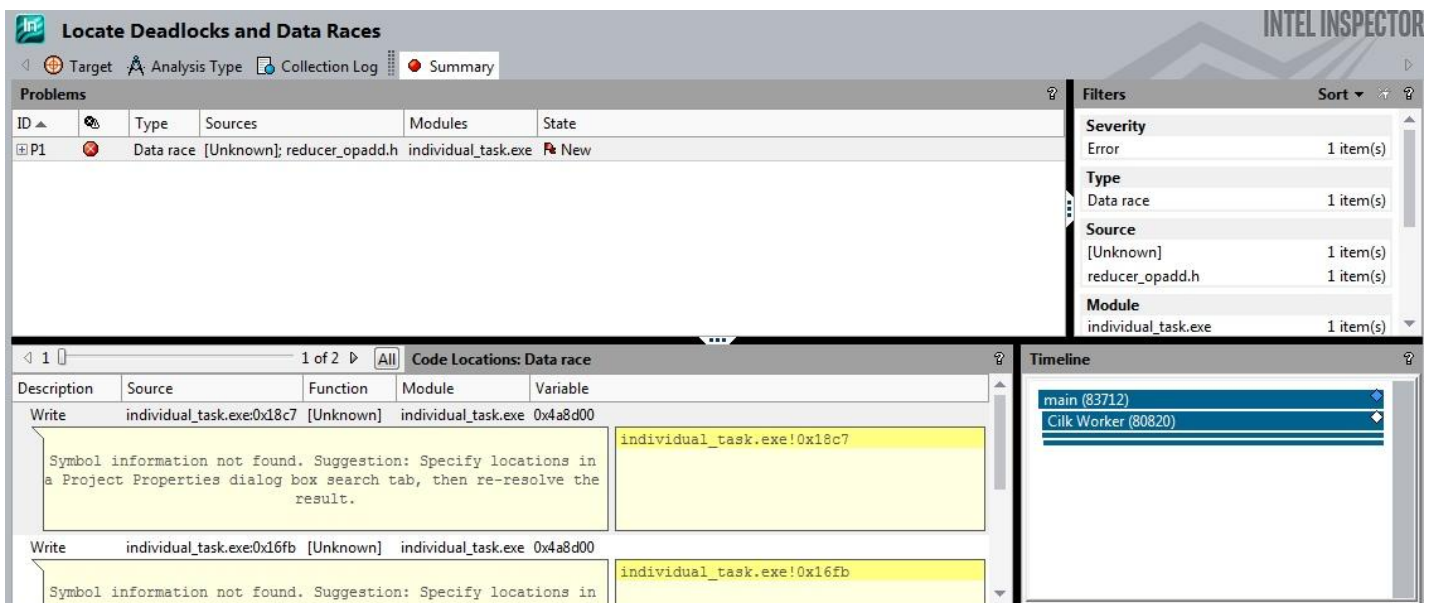
Parallel integral = 6.28319
Parallel time = 0.129915

Acceleration = 1.86803

D:\Мои документы\Maria\individual_task\individual_task\Release\individual_task.exe
Процесс 77044 завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу:
```

Получили ускорение процесса в 1.86 раз.

Проверим параллельную реализацию с помощью *Intel Parallel Inspector XE*:



**Locate Deadlocks and Data Races**

Target Analysis Type Collection Log Summary

ID	Type	Sources	Modules	State
P1	Data race	[Unknown]; reducer_opadd.h	individual_task.exe	New

**Filters**

Severity	Count
Error	1 item(s)

**Type**

Type	Count
Data race	1 item(s)

**Source**

Source	Count
[Unknown]	1 item(s)
reducer_opadd.h	1 item(s)

**Module**

Module	Count
individual_task.exe	1 item(s)

**Code Locations: Data race**

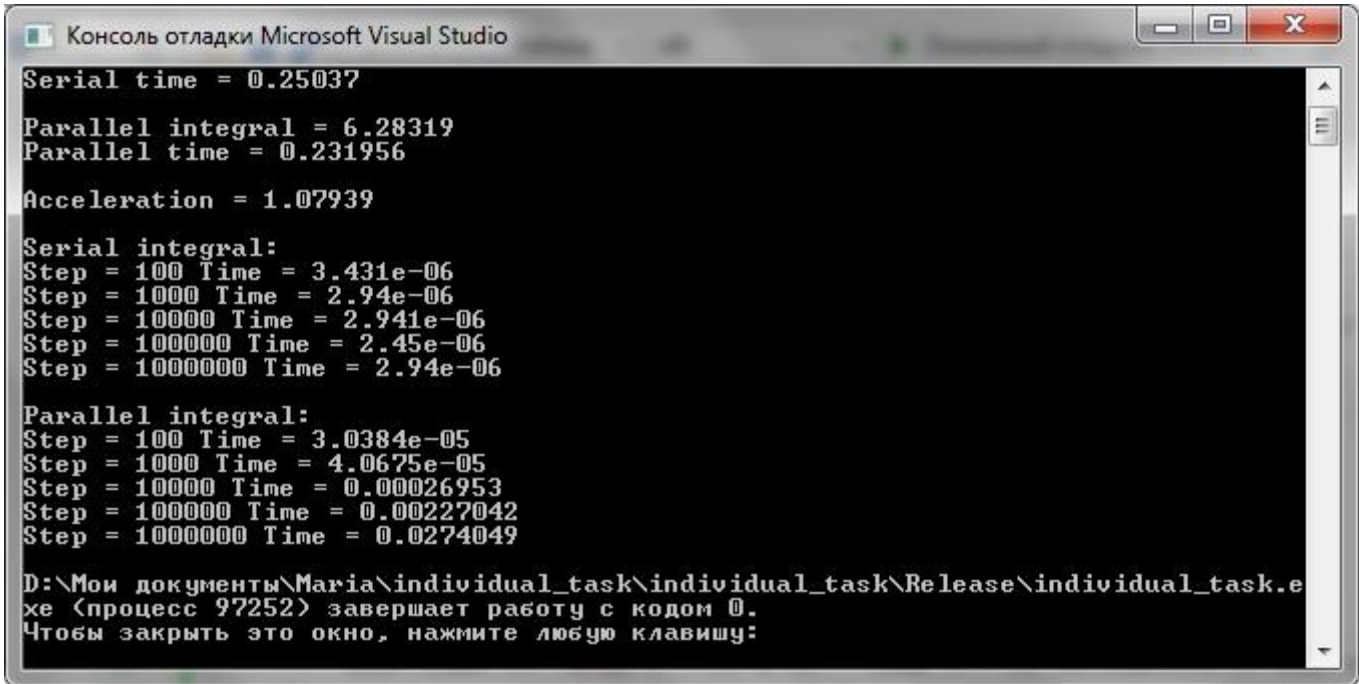
Description	Source	Function	Module	Variable
Write	individual_task.exe:0x18c7	[Unknown]	individual_task.exe	0x4a8d00
Symbol information not found. Suggestion: Specify locations in a Project Properties dialog box search tab, then re-resolve the result.				
individual_task.exe!0x18c7				
Write	individual_task.exe:0x16fb	[Unknown]	individual_task.exe	0x4a8d00
Symbol information not found. Suggestion: Specify locations in				
individual_task.exe!0x16fb				

**Timeline**

Thread	Address
main (83712)	
Cilk Worker (80820)	

Ошибок не обнаружено.

Далее: меняя величину шага разбиения, посмотрим на скорость выполнения:



```
Консоль отладки Microsoft Visual Studio

Serial time = 0.25037

Parallel integral = 6.28319
Parallel time = 0.231956

Acceleration = 1.07939

Serial integral:
Step = 100 Time = 3.431e-06
Step = 1000 Time = 2.94e-06
Step = 10000 Time = 2.941e-06
Step = 100000 Time = 2.45e-06
Step = 1000000 Time = 2.94e-06

Parallel integral:
Step = 100 Time = 3.0384e-05
Step = 1000 Time = 4.0675e-05
Step = 10000 Time = 0.00026953
Step = 100000 Time = 0.00227042
Step = 1000000 Time = 0.0274049

D:\Мои документы\Maria\individual_task\individual_task\Release\individual_task.exe (процесс 97252) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу:
```

Как видим, для меньшего числа отрезков разбиения параллельный метод работает лучше.