Индивидуальное задание

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

$$\int_0^1 \frac{8}{\sqrt{1+x^2}}dx$$

В математическом анализе обосновывается аналитический способ нахождения значения интеграла с помощью формулы Ньютона-Лейбница

$$\int_a^b f(x)dx = F(b) - F(a)$$

Однако применение данного подхода к вычислению наталкивается на несколько серьезных препятствий:

- Для многих функций не существует первообразной среди элементарных функций;
- Даже если первообразная для заданной функции существует, то вычисление двух ее значений $F(a)$, $F(b)$ может оказаться более трудоемким, чем вычисление существенно большего количества значений $f(x)$;
- Для многих реальных приложений определенного интеграла характерная дискретность задания подынтегральной функции, что делает аналитический подход неприменимым.

Поэтому необходимо использовать приближенные формулы для вычисления определенного интеграла на основе значений подынтегральной функции. Такие специальные формулы называются квадратурными формулами или формулами численного интегрирования.

В качестве инструментов параллелизации, используемых для решения задачи, мы будем использовать:

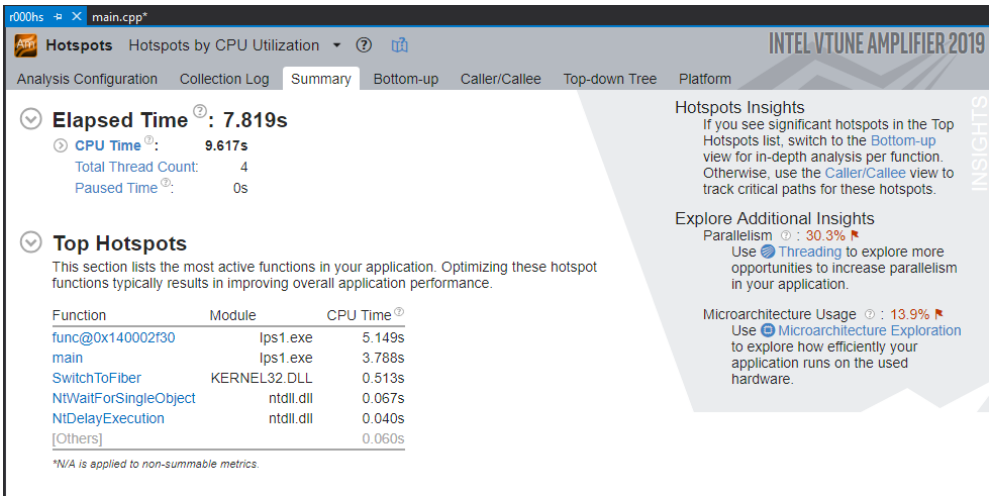
- Cilk Plus – это расширения языка C/C++, которое помогает с введением параллелизма в код программы;
- Intel Parallel Studio XE – это набор программных продуктов от компании Intel;
    1. Intel Parallel Inspector – инструмент, предназначенный для тестирования работающей программы с целью выявления основных ошибок, которые возникают при разработке параллельного кода;

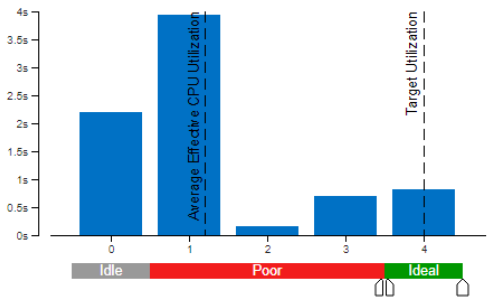
2. Intel Amplifier – инструмент, который используется для профилирования приложения с целью выявления наиболее часто используемых участков программы (hotspots), а также узких мест (bottleneck) в работе программы. Этот инструмент также позволяет анализировать параллельные программы на эффективность использования ими ресурсов процессора;

Вычисление интеграла с помощью формулы средних прямоугольников

```
//! middle rectangles
//! Sequential Integral Calculation
double sequential_intergal(int n)
{
	double sum = 0.0;
	double h = (b - a) / (n - 1);
	for (auto i = 0; i < n; i++)
		sum += func(i * h) + func((i + 1)*h);
	return sum * h / 2.0;
}
```

Amplifier XE: определяем наиболее часто используемые участки кода





Collection and Platform Info

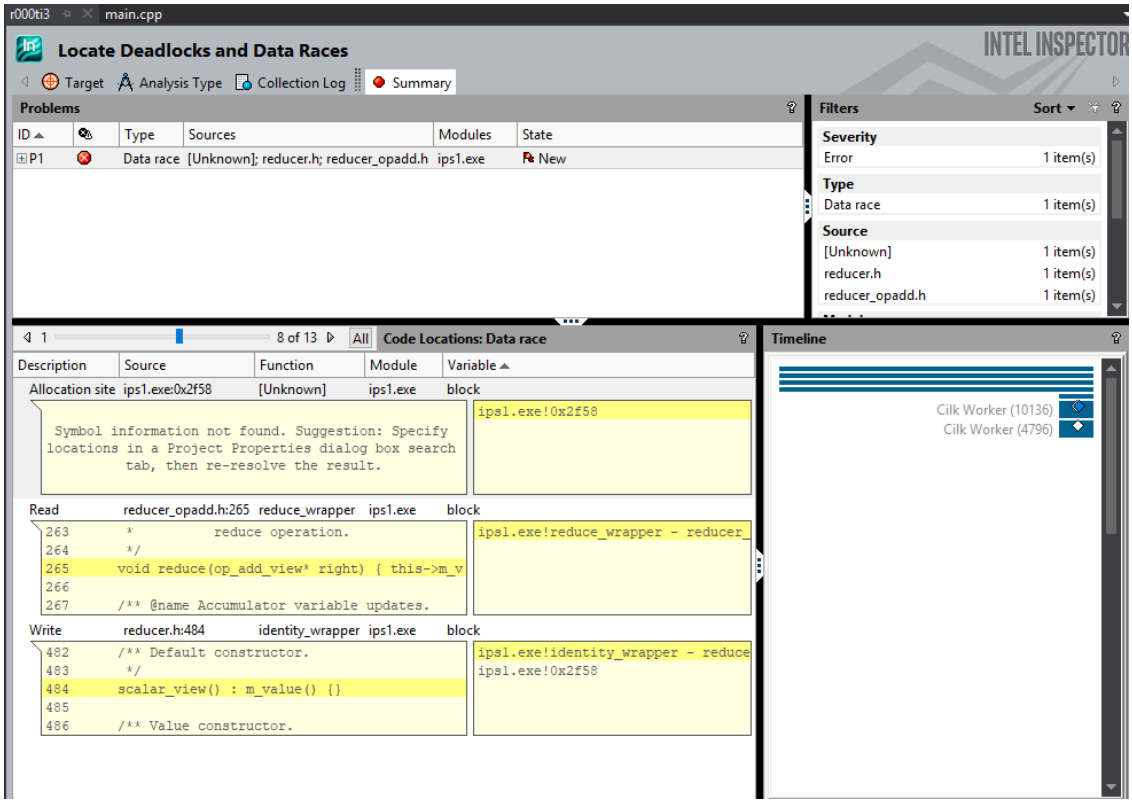
This section provides information about this collection, including result set size and collection platform data.

Application Command Line:	C:\Users\Vadim\source\repos\lps1\x64\Release\lps1.exe
Environment Variables:	
Operating System:	Microsoft Windows 10
Computer Name:	DESKTOP-RC2B2HV
Result Size:	3 MB
Collection start time:	18:45:09 28/11/2019 UTC
Collection stop time:	18:45:17 28/11/2019 UTC
Collector Type:	Event-based counting driver,User-mode sampling and tracing

Параллельное вычисление интеграла

```
//! Parallel Integral Calculation
double parallel_integral(int n)
{
	double h = (b - a) / n;
	cilk::reducer_opadd<double> sum(0.0);
	cilk_for (auto i = 0; i < n; i++)
	{
		sum += func(i * h) + func((i + 1)*h);
	}
	return sum->get_value()*h / 2.0;
}
```

Inspector XE: определяем данные, которые принимают участие в гонке данных или в других основных ошибках, возникающих при разработке параллельных программ



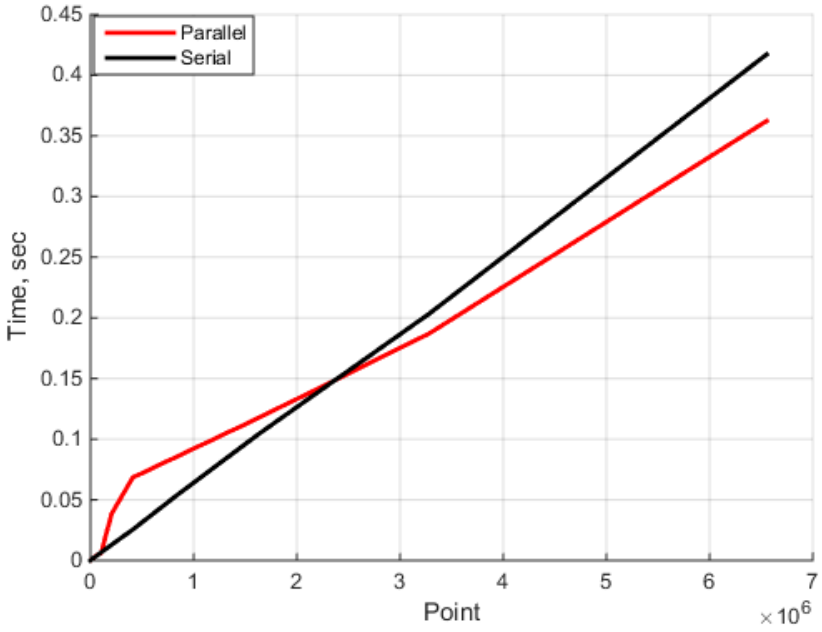
Гонка данных в reducer.h

В моей программе ошибок не обнаружено.

Запустим полученную программу

```
Integral = 7.05099
Elapsed time = 5.93052
Parallel Integral = 7.05099
Elapsed time = 3.336
Boost = 1.77773
Для продолжения нажмите любую клавишу . . .
```

Построим график зависимости времени выполнения от заданных параметров алгоритма.



При увеличении количества точек разбиения, параллельная реализация работает быстрее, чем последовательная.