Индивидуальное задание

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

$$\int_0^1 \frac{4}{\sqrt{4-x^2}}\,dx$$

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.

Сказанное предопределяет необходимость использования приближенных формул 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;

Реализуем вычисление интеграла с помощью формулы правых прямоугольников.

```
double integrate(double a, double b, size_t N)
{
    const double h = (b - a) / N;
    double sum = 0.0f;
    for (size_t i = 0; i < N; ++i)
        sum += fun(a + i * h);

    return sum * h;
}
```

С помощью инструмента Amplifier XE определим наиболее часто используемые участки кода.

Top Hotspots

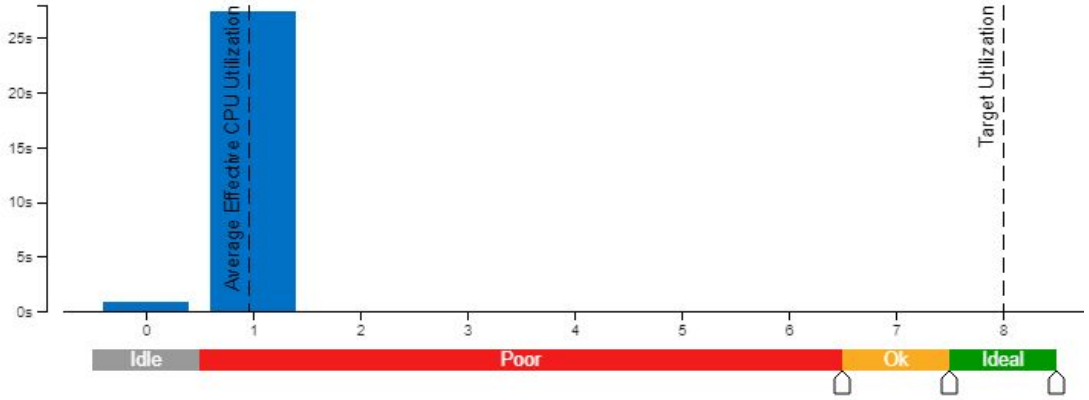
This section lists the most active functions in your application. Optimizing these hotspot functions typically results in impro

Function	Module	CPU Time
fun	IPS_labs.exe	11.644s
integrate	IPS_labs.exe	9.061s
_libm_sqrt_ex	libmmdd.dll	5.481s
sqrt	libmmdd.dll	1.009s
[Import thunk sqrt]	IPS_labs.exe	0.020s
[Others]		0.013s

**N/A is applied to non-summable metrics.*

Effective CPU Utilization Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin an



Распараллелим подсчет суммы:

```
double parallel_integrate(double a, double b, size_t n)
{
    const double h = (b - a) / n;
    cilk::reducer_opadd<double> sum(0.0);
    cilk_for(size_t i = 0; i < n; ++i)
        sum += fun(a + i * h);

    return sum->get_value() * h;
}
```

Top Hotspots

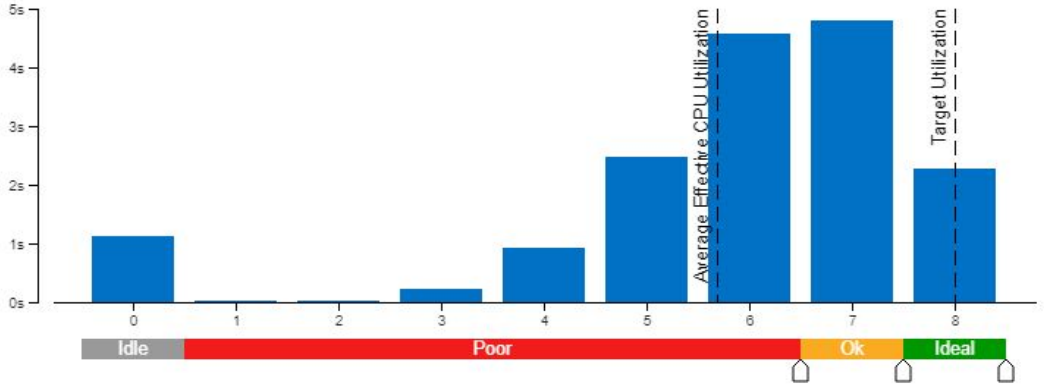
This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving

Function	Module	CPU Time
func@0x140005085	IPS_labs.exe	19.943s
cilk::internal::reducer_base<struct cilk::op_add<double,1> >::view	IPS_labs.exe	17.733s
[Cilk reducer]	cilkrts20.dll	15.018s
fun	IPS_labs.exe	14.503s
cilk::op_add_view<double>::operator+=	IPS_labs.exe	11.039s
[Others]		34.452s

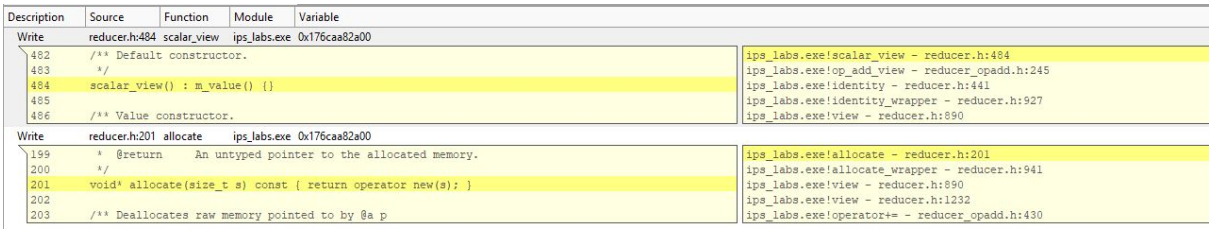
*N/A is applied to non-summable metrics.

Effective CPU Utilization Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Ov

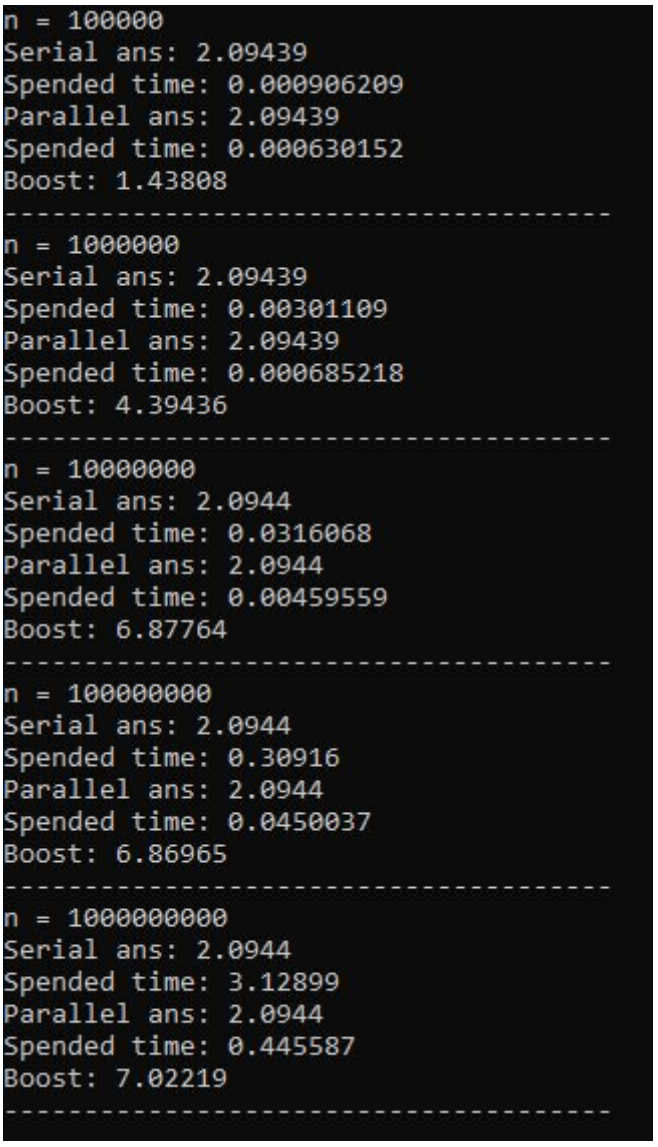


Далее, используя Inspector XE, определим те данные, которые принимают участие в гонке данных или в других основных ошибках, возникающих при разработке параллельных программ.



Ошибок не обнаружено.

Скомпилируем программу в режиме Release и запустим для различных значений N



Как видим, ответ во всех случаях корректный, а скорость вычислений выше в параллельной реализации, причем выигрыш увеличивается пропорционально N.