

---

# Multicore Computing Project4

---

- Problem 2 -



---

과목명	멀티코어컴퓨팅
제출일	2023.06.12.
학 번	20183901
학 과	소프트웨어학부
이 름	김상민

---

# 목 차

---

1. Program setup
  2. Source code
  3. Program results
  4. Explanation
-

## 1. Program setup

Execution environment: Google Colab

GPU: Python 3 Google Compute Engine 백엔드

## 2. Source code

```
#include <thrust/device_vector.h>
#include <thrust/functional.h>
#include <thrust/transform_reduce.h>
#include <thrust/execution_policy.h>
#include <iostream>

struct calculate_sum : public thrust::unary_function<long, double>
{
    double step;

    calculate_sum(double step) : step(step) {}

    __device__
    double operator()(const long& i) const
    {
        double x = (i + 0.5) * step;
        return 4.0 / (1.0 + x * x);
    }
};

int main()
{
    long num_steps = 1000000000;
    double step = 1.0 / static_cast<double>(num_steps);

    thrust::device_vector<long> indices(num_steps);
    thrust::sequence(indices.begin(), indices.end());

    clock_t start_time = clock();

    double sum = thrust::transform_reduce(thrust::device, indices.begin(), indices.end(),
                                           calculate_sum(step), 0.0, thrust::plus<double>());

    double pi = step * sum;

    clock_t end_time = clock();

    clock_t diff_time = end_time - start_time;
    printf("execution time: %f sec. \n", (double)diff_time/CLOCKS_PER_SEC);

    printf("pi = %.10lf\n", pi);

    return 0;
}
```

### 3. Program results

#### OpenMP - 1

```
!./a.out
```

```
Execution Time : 7.9901780330sec  
pi=3.1415926536
```

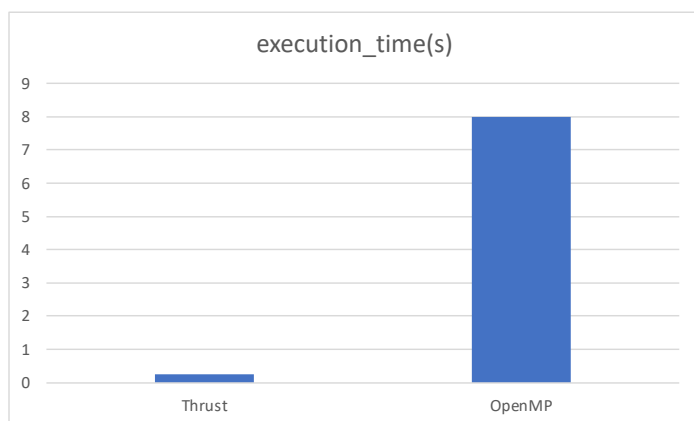
#### Thrust

```
!./a.out
```

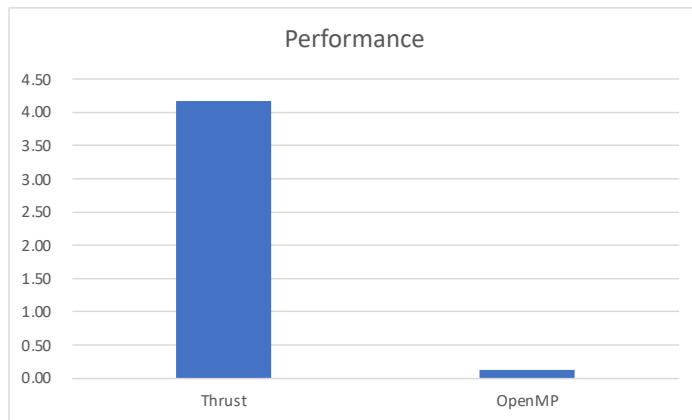
```
execution time: 0.248386 sec.  
pi = 3.1415926536
```

#### Result

	Thrust	OpenMP
execution_time(s)	0.24	8



	Thrust	OpenMP
Performance	4.17	0.13



#### 4. Explanation

Thrust is a C++ template library for CUDA based on the Standard Template Library. Thrust allows you to implement high performance in parallel. If we look at the result, Thrust implemented version shows outstanding performance compared to OpenMp(threads: 1) version. This shows that Program run on GPU performs much better than on CPU.