2023-1 Multicore Computing, Project #4

Problem 2

2023-05-24

CAU SW 20184286

LEE DONGHWA

Environment

Google Colab (GPU type : T4)

Just run the Jupyter file in Colab.

Compile > !nvcc {filename}

Execute > !./a.out

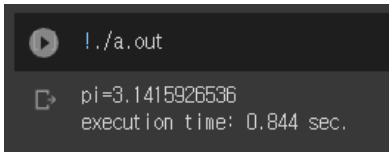
Source code

Thrust_ex.cu

```
%%writefile thrust_ex.cu
#include <thrust/host_vector.h>
#include <thrust/device_vector.h>

#include <thrust/transform.h>
#include <thrust/sequence.h>

#include <stdio.h>
#include <time.h>
#include <iostream>

long num_steps = 200000;
double step = 1.0/(double) num_steps;

template<typename T>
struct integral
{
    double step;
```

```
    integral(double step) : step(step){}

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

int main(){
    clock_t start_time = clock();
    thrust::device_vector<int> index(num_steps);

    // make new sequence
    thrust::sequence(index.begin(), index.end());

    integral<double> unary_op(step);
    thrust::plus<double> binary_op;
    double init = 0.0;

    // transform (using function unary_op, set by 'integral') and
reduction (reduce to a single value, set by summation)
    double sum = thrust::transform_reduce(index.begin(), index.end(),
unary_op, init, binary_op);
    double pi = step * sum;
    printf("pi=%.10lf\n",pi);

    clock_t end_time = clock();

    clock_t diff_time = end_time - start_time;
    printf("execution time: %.3lf sec. \n",
(double)diff_time/CLOCKS_PER_SEC);

    return 0;
}
```

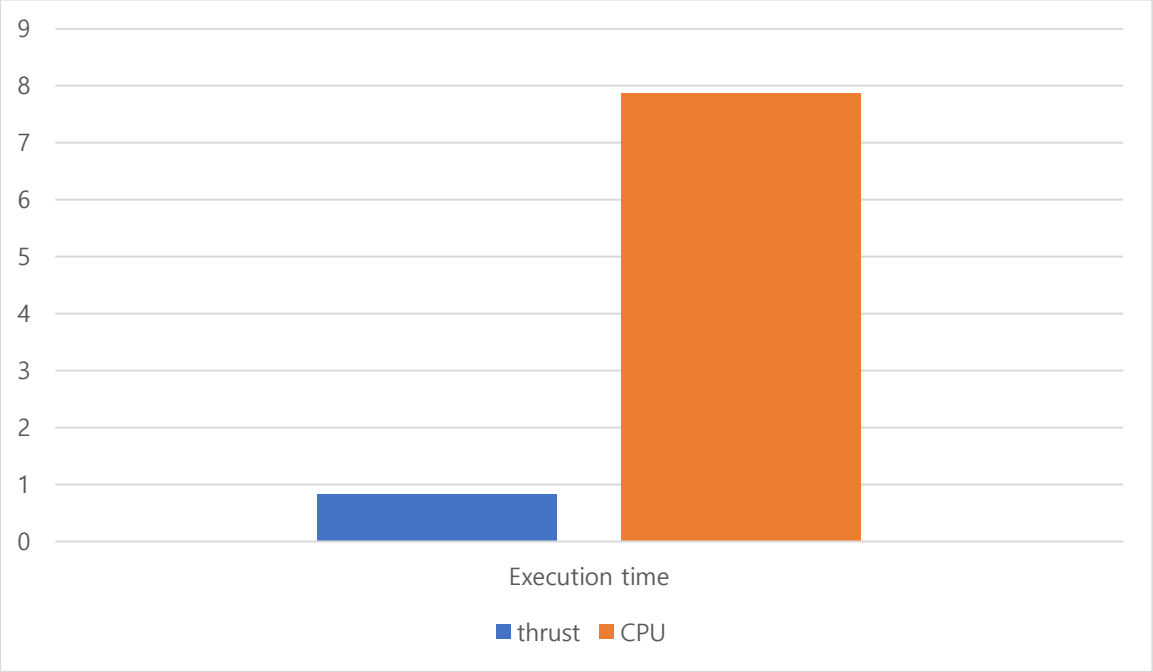
Result



```
[51]  !nvcc omp_pi_one.c
[52]  !./a.out
      execution time: 7.871 sec.
      pi=3.1415926536
```

Tables_ (unit : sec)

	Execution time
thrust	0.844
CPU	7.871



Explanation / Analysis_

When using the CPU for calculations, I found that using Thrust resulted in significantly shorter execution times compared to the CPU alone. This advantage was also evident in integration calculations, where using the GPU for computation proved to be more favorable than relying on the CPU. Through experimentation, I was able to confirm these findings.