

Домашнее задание 4

Корнилов Георгий, вариант 11

ФКН ПИ НИУ ВШЭ, группа БПИ-197

1 Текст задачи

Определить индексы $i, j (i \neq j)$, для которых выражение $A[i] - A[i + 1] + A[i + 2] - A[i + 3] + \dots \pm A[j]$ имеет максимальное значение. Входные данные: массив чисел A , произвольной длины большей 10. Количество потоков не является параметром задачи.

2 Методы решения задачи

Воспользуемся парадигмой итеративного параллелизма, где «цикл» по i идет от 0 до $n - 2$.

Внутри каждого потока идет цикл по j от $i + 1$ до $n - 1$, в котором считается сумма $A[i] - A[i + 1] + A[i + 2] - A[i + 3] + \dots \pm A[j]$ и обновляется максимальный индекс j для данного индекса i с значением максимальной суммы потока *cur_max_sum*.

После выполнения цикла обновляется значение максимальной суммы *max_sum* значением *cur_max_sum*.

3 Исходный код программы

```
#include <iostream>
#include <ctime>
#include <iomanip>
#include <fstream>
#include <thread>
#include <mutex>

std::mutex arr_mutex;
std::mutex max_mutex;

size_t arr_size;
int *arr;
int max_i = -1, max_j;
```

```

long long max_sum;

void func(int startIndex) {
    int max_index = startIndex + 1;
    arr_mutex.lock();
    const int temp_size = arr_size;
    arr_mutex.unlock();

    arr_mutex.lock();
    long long cur_sum = arr[startIndex] - arr[startIndex + 1];
    arr_mutex.unlock();
    long long cur_max_sum = cur_sum;

    for (auto offset = 2; startIndex + offset < temp_size; offset++) {
        arr_mutex.lock();
        int elem = arr[startIndex + offset];
        arr_mutex.unlock();

        if (offset % 2 == 0) {
            cur_sum += elem;
        }
        else {
            cur_sum -= elem;
        }

        if (cur_sum >= cur_max_sum) {
            cur_max_sum = cur_sum;
            max_index = startIndex + offset;
        }
    }

    max_mutex.lock();
    if (max_i == -1 || cur_max_sum > max_sum) {
        max_i = startIndex;
        max_j = max_index;
        max_sum = cur_max_sum;
    }
    max_mutex.unlock();
}

int main(int argc, char **argv) {
    if (argc != 2)
    {
        std::cout << "Wrong usage: main.exe <input_path>" << std::endl;
    }
    char *file_name = argv[1];

```

```

std::ifstream input(file_name);
if (!input.is_open()) {
    std::cout << "wrong file" << std::endl;
    return 1;
}
input >> arr_size;

if (arr_size <= 10) {
    std::cout << "Incorrect size of vector = " << arr_size << std::endl;
    return 1;
}

arr = new int[arr_size];

try {
    for (size_t i = 0; i < arr_size; i++) {
        input >> arr[i];
    }
}
catch (...)
{
    std::cout << "wrong content of file" << std::endl;
    input.close();
    return 1;
}
input.close();

std::cout << arr_size << std::endl;
for (size_t i = 0; i < arr_size; i++) {
    std::cout << arr[i] << " ";
}
std::cout << std::endl;

std::thread *threads = new std::thread[arr_size - 1];

clock_t startTime = clock();

for (size_t i = 0; i < arr_size - 1; i++) {
    threads[i] = std::thread(func, i);
}

for (size_t i = 0; i < arr_size - 1; i++) {
    threads[i].join();
}

```

```

        clock_t endTime = clock();
        std::cout << "Max sum occurs with i = " << max_i + 1 << ", j = " << max_j + 1 << " and is equal to " << maxSum << "\n";
        std::cout << "Calculation time = " << 1.0 * (endTime - startTime) / CLOCKS_PER_SEC << " seconds\n";

        delete[] arr;
        delete[] threads;
        return 0;
    }
}

```

4 Примеры работы программы

Примеры работы программы на разных тестовых данных (тестовые данные приложены по [ссылке](#)):

```

C:\Users\Mi\Desktop\architectures\testhw3>main.exe test1.txt
12
1 2 -1 4 5 -10 2 4 2 -100 1 2
Max sum occurs with i = 4, j = 10 and is equal to 116
Calculation time = 0.004 seconds

C:\Users\Mi\Desktop\architectures\testhw3>main.exe test2.txt
11
-10 1 2 -3 4 -5 4 -2 3 -1 1
Max sum occurs with i = 2, j = 10 and is equal to 25
Calculation time = 0.002 seconds

C:\Users\Mi\Desktop\architectures\testhw3>main.exe test3.txt
Incorrect size of vector = 5

```

5 Источники информации

1. <http://www.softcraft.ru/edu/comparch/practice/thread/02-sync/>
2. <https://pro-prof.com/forums/topic/parallel-programming-paradigms>