

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

**ПРОЕКТ ПО ДИСЦИПЛИНЕ «АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ»
ПО МНОГОПОТОЧНОСТИ НА C++
Отчет**

Исполнитель
Студентка группы БПИ193(подгруппа 1)
Шевко Марина Николаевна
Вариант 26
16 ноября 2020 г

Москва 2020

СОДЕРЖАНИЕ

| | |
|--|-----------|
| 1. ВВЕДЕНИЕ | 3 |
| 1.1. Постановка задачи на выполнение | 3 |
| 2. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ АЛГОРИТМОВ | 4 |
| 2.1. Модель взаимодействия..... | 4 |
| 2.2. Алгоритм | 4 |
| 2.2.1. Портфель задач | 4 |
| 2.2.2. Считывание входных данных и заполнение секторов острова | 4 |
| 2.2.3. Функция поиска клада | 5 |
| 3. ТЕСТИРОВАНИЕ ПРОГРАММЫ | 7 |
| 3.1. Тестирование обработки неверных входных данных | 7 |
| 3.2. Тестирование распределения задач по потокам | 7 |
| 3.2.1. Тестирование на небольшом числе. | 7 |
| 3.2.2. Тестирование на большем числе | 9 |
| 4. ТЕКСТ ПРОГРАММЫ | 12 |
| 5. СПИСОК ИСТОЧНИКОВ | 15 |

1. ВВЕДЕНИЕ

1.1. Постановка задачи на выполнение

Условие (Вариант 26):

Вторая задача об Острове Сокровищ. Шайка пиратов под предводительством Джона Сильвера высадилась на берег Острова Сокровищ. Не смотря на добытую карту старого Флинта, местоположение сокровищ по прежнему остается загадкой, поэтому искать клад приходится практически на ощупь. Так как Сильвер ходит на деревянной ноге, то самому бродить по джунглям ему не с руки. Джон Сильвер поделил остров на участки, а пиратов на небольшие группы. Каждой группе поручается искать клад на нескольких участках, а сам Сильвер ждет на берегу. Группа пиратов, обшарив одну часть острова, переходит к другой, еще необследованной части. Закончив поиски, пираты возвращаются к Сильверу и докладывают о результатах. Требуется создать многопоточное приложение с управляющим потоком, моделирующее действия Сильвера и пиратов. При решении использовать парадигму портфеля задач.

2. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ АЛГОРИТМОВ

2.1. Модель взаимодействия

По условию задачи существует управляющий поток – Сильвер, дающий задачи пиратам на обыск сегмента острова и обрабатывающий их результаты выполнения, и рабочие процессы - группы пиратов. По условию, нужно использовать парадигму портфеля задач, поэтому в качестве модели взаимодействия используется Управляющий - рабочие (распределенный портфель задач). Соответственно, в портфеле задач мы храним задачи по обработке одной области острова. Управляющий (Сильвер) дает каждой из групп пиратов задачу из портфеля задач (представлен распределенной переменной, к которой могут обращаться потоки) по поиску клада на определенном секторе острова. Как только какая-нибудь из групп заканчивает работу и сообщает капитану Сильверу о результатах, она берет следующую, еще не выполненную задачу, которую еще не взяла на выполнение другая группа.

2.2. Алгоритм

Реализовано с помощью библиотеки <thread>.

2.2.1. Портфель задач

Портфель задач представляет собой массив из структур Sector, которые хранят данные о задаче – номер обыскиваемого сектора, его размер и находится ли на нем клад.

```
struct Sector
{
    int size = 0;
    bool withMoney = false;
};

Sector* sectors; // портфель задач
```

2.2.2. Считывание входных данных и заполнение секторов острова

Пользователь вводит число секторов на острове: целое число от 0 до 1000.

```
//Запрос числа секторов от пользователя
int getNoOfSectors()
{
    cout << "Input number of search sectors on the area (0 < x < 1000): ";
    int num;
    cin >> num;
    while (num <= 0 or num > 1000)
    {
        printf("Incorrect input! Try again: ");
        cin >> num;
    }
    return num;
}
```

Рандомно заполняются размеры секторов и выбирается сектор кладом

```

//инициализируем секторы острова
sectors = new Sector[sectorsNum];

// генерируем сектора острова - у каждого свой размер.
for (int i = 0; i < sectorsNum; i++)
{
    sectors[i] = Sector{ rand(), false };
}

// случайно выбираем сектор с кладом
int sectorId = (double)rand() / RAND_MAX * sectorsNum;
sectors[sectorId].withMoney = true;

```

У пользователя запрашивается целое число поисковых групп пиратов – от 0 до максимального количества одновременных потоков, поддерживаемых системой (кол-во логических ядер).

```

//Запрос числа групп от пользователя
int getNoOfGroups()
{
    // Кол-во одновременных потоков, поддерживаемых системой
    unsigned int n = std::thread::hardware_concurrency();

    printf("Please input number of groups (0 < x < %d): ", n);

    int num;
    cin >> num;
    while (num <= 0 or num > n)
    {
        printf("Incorrect input! Try again: ");
        cin >> num;
    }
    return num;
}

```

Создается кол-во потоков равное кол-во поисковых групп пиратов. Они будут симулировать их действия, выполняющие функцию поиска клада.

```

// создание потоков
printf("----- Silver's report-----\n");
auto t_start = std::chrono::high_resolution_clock::now();
for (int i = 0; i < threadNumber; i++)
{
    thr[i] = thread(findMoney, i);
}

```

2.2.3. Функция поиска клада

Сильвер дает группе пиратов задачу из портфеля задач на обыск сегмента, который на данный момент не обследован. На этот момент доступ портфелю блокируется для других групп (потоков), чтобы группа не взяла задачу, которая находится в процессе выполнения другой группой. В консоль выводится информация о том, что группа с соответствующим названием взяла задачу по обследованию i-го сектора.

```

//ищем пока клад не найден
while (moneyFound)
{
    //Сильвер берет сектор из списка секторов и выдает задание группе Name
    //блокируем для других групп
    mtx.lock();
    //Сильвер дает сектор i текущей группе и увеличивает счетчик
    int i = currentSector++;
    printf("---> Pirates group %d has got sector No.%d \n", name, i);
    mtx.unlock();
}

```

Далее происходит проверка сектора группой пиратов. Для вывода времени поиска засекается время в начале и в конце операции.

```

//Проверка секторов - занимает разное время
auto t_start = std::chrono::high_resolution_clock::now();
double d = uselessFunc(sectors[i].size); // нагружаем группу объемом работы, зависящим от размера поискового сектора
auto t_end = std::chrono::high_resolution_clock::now();
double dt = std::chrono::duration<double, std::milli>(t_end - t_start).count();

```

Для симуляции долговременной и ресурсозатратной операции поиска клада на сегменте используется uselessFunc. Чем больше размер (size), тем дольше она будет выполняться. Используется именно реальная функция, а не Sleep() потока, чтобы далее проследить нагрузку на ядра процессора.

```

// Время- и ресурсно- затратная функция
double uselessFunc(const int size)
{
    double d = 1;
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            d += d * i * j - d;
        }
    }
    return d;
}

```

Выводим сообщение в консоль о результатах поисков (пираты возвращаются к Сильверу с результатами поисков), блокируем поток на вывод, чтобы не было конфликтов. Если клад был найден, то изменяется глобальная переменная moneyFound и все потоки завершают свои поиски (свободные - не берут новые задачи, занятые - заканчивают текущую, сообщают результаты Сильверу и тоже больше не берут).

```

if (sectors[i].withMoney)
{
    // блокируем поток на вывод в консоль
    mtx.lock();
    printf("!!!! Pirates group %d has found money in the sector No.%d, Elapsed time %f mcs \n", name, i, dt);
    mtx.unlock();
    moneyFound = false;
    group = name;
}
else
{
    mtx.lock();
    printf("<<<< Pirates group %d has NOT found money in the sector No.%d, Spent time %f mcs for %d size (%f)\n", name, i, dt, sectors[i].size, d);
    mtx.unlock();
}

```

3. ТЕСТИРОВАНИЕ ПРОГРАММЫ

3.1. Тестирование обработки неверных входных данных

```
Input number of search sectors on the area (0 < x < 1000): 1001
Incorrect input! Try again: -2
Incorrect input! Try again: 12312312
Incorrect input! Try again: 500
INFO: Money has been located in the sector No.19

Please input number of groups (0 < x <= 16): 20
Incorrect input! Try again: -1
Incorrect input! Try again: 12312
Incorrect input! Try again: 5
----- Silver's report-----
---> Pirates group 0 has got sector No.0
---> Pirates group 1 has got sector No.1
```

Просим пользователя вводить данных до тех пор, пока они не будут валидными.

3.2. Тестирование распределения задач по потокам

3.2.1. Тестирование на небольшом числе.

Пользователь задает 100 секторов на острове и выделяет 4 группы пиратов на поиски (из доступных 16). Клад случайно помещается в 14 сектор и находится верно. Освободившийся поток действительно берет следующую задачу из портфеля задач. Т.к. задачи - поиск по сегменту острова – разные по размеру, потоки тратят разное время на их выполнение.

```
Input number of search sectors on the area (0 < x < 1000): 100
INFO: Money has been located in the sector No.14

Please input number of groups (0 < x < 16): 4
----- Silver's report-----
---> Pirates group 0 has got sector No.0
---> Pirates group 2 has got sector No.1
<<<< Pirates group 0 has NOT found money in the sector No.0, Spent time 0.032000 mcs for 41 size (0.000000)
---> Pirates group 0 has got sector No.2
---> Pirates group 1 has got sector No.3
---> Pirates group 3 has got sector No.4
<<<< Pirates group 0 has NOT found money in the sector No.2, Spent time 141.215300 mcs for 6334 size (0.000000)
---> Pirates group 0 has got sector No.5
<<<< Pirates group 0 has NOT found money in the sector No.5, Spent time 862.487400 mcs for 15724 size (0.000000)
---> Pirates group 0 has got sector No.6
<<<< Pirates group 2 has NOT found money in the sector No.1, Spent time 1191.438100 mcs for 18467 size (0.000000)
---> Pirates group 2 has got sector No.7
<<<< Pirates group 3 has NOT found money in the sector No.4, Spent time 1287.227400 mcs for 19169 size (0.000000)
---> Pirates group 3 has got sector No.8
<<<< Pirates group 0 has NOT found money in the sector No.6, Spent time 459.653400 mcs for 11478 size (0.000000)
---> Pirates group 0 has got sector No.9
<<<< Pirates group 1 has NOT found money in the sector No.3, Spent time 2451.371300 mcs for 26500 size (0.000000)
---> Pirates group 1 has got sector No.10
<<<< Pirates group 1 has NOT found money in the sector No.10, Spent time 113.548600 mcs for 5705 size (0.000000)
---> Pirates group 1 has got sector No.11
<<<< Pirates group 0 has NOT found money in the sector No.9, Spent time 2090.449400 mcs for 24464 size (0.000000)
---> Pirates group 0 has got sector No.12
<<<< Pirates group 3 has NOT found money in the sector No.8, Spent time 2537.570500 mcs for 26962 size (0.000000)
---> Pirates group 3 has got sector No.13
<<<< Pirates group 2 has NOT found money in the sector No.7, Spent time 3007.445700 mcs for 29358 size (0.000000)
---> Pirates group 2 has got sector No.14
!!!! Pirates group 2 has found money in the sector No.14, Elapsed time 346.151300 mcs
<<<< Pirates group 3 has NOT found money in the sector No.13, Spent time 990.603500 mcs for 16827 size (0.000000)
<<<< Pirates group 1 has NOT found money in the sector No.11, Spent time 2757.605800 mcs for 28145 size (0.000000)
<<<< Pirates group 0 has NOT found money in the sector No.12, Spent time 1884.180000 mcs for 23281 size (0.000000)

RESULTS: Pirates group 2 has found Money in the sector No.14 for 5441.568500 ms
```

Далее протестируем с тем же количеством секторов (100), но с использованием 8 потоков. Результат так же верен.

```

Microsoft Visual Studio Debug Console

Input number of search sectors on the area (0 < x < 1000): 100
INFO: Money has been located in the sector No.14

Please input number of groups (0 < x < 16): 8
----- Silver's report-----
---> Pirates group 3 has got sector No.0
<<<< Pirates group 3 has NOT found money in the sector No.0, Spent time 0.009800 mcs for 41 size (0.000000)
---> Pirates group 3 has got sector No.1
---> Pirates group 1 has got sector No.2
---> Pirates group 4 has got sector No.3
---> Pirates group 0 has got sector No.4
---> Pirates group 5 has got sector No.5
---> Pirates group 6 has got sector No.6
---> Pirates group 2 has got sector No.7
---> Pirates group 7 has got sector No.8
<<<< Pirates group 1 has NOT found money in the sector No.2, Spent time 140.030400 mcs for 6334 size (0.000000)
---> Pirates group 1 has got sector No.9
<<<< Pirates group 6 has NOT found money in the sector No.6, Spent time 459.893900 mcs for 11478 size (0.000000)
---> Pirates group 6 has got sector No.10
<<<< Pirates group 6 has NOT found money in the sector No.10, Spent time 113.638900 mcs for 5705 size (0.000000)
---> Pirates group 6 has got sector No.11
<<<< Pirates group 5 has NOT found money in the sector No.5, Spent time 864.077400 mcs for 15724 size (0.000000)
---> Pirates group 5 has got sector No.12
<<<< Pirates group 3 has NOT found money in the sector No.1, Spent time 1193.485900 mcs for 18467 size (0.000000)
---> Pirates group 3 has got sector No.13
<<<< Pirates group 0 has NOT found money in the sector No.4, Spent time 1283.135800 mcs for 19169 size (0.000000)
---> Pirates group 0 has got sector No.14
!!!! Pirates group 0 has found money in the sector No.14, Elapsed time 346.350600 mcs
<<<< Pirates group 3 has NOT found money in the sector No.13, Spent time 988.551800 mcs for 16827 size (0.000000)
<<<< Pirates group 1 has NOT found money in the sector No.9, Spent time 2092.377500 mcs for 24464 size (0.000000)
<<<< Pirates group 4 has NOT found money in the sector No.3, Spent time 2452.966900 mcs for 26500 size (0.000000)
<<<< Pirates group 7 has NOT found money in the sector No.8, Spent time 2538.605400 mcs for 26962 size (0.000000)
<<<< Pirates group 5 has NOT found money in the sector No.12, Spent time 1893.008700 mcs for 23281 size (0.000000)
<<<< Pirates group 2 has NOT found money in the sector No.7, Spent time 3007.882500 mcs for 29358 size (0.000000)
<<<< Pirates group 6 has NOT found money in the sector No.11, Spent time 2757.959700 mcs for 28145 size (0.000000)

RESULTS: Pirates group 0 has found Money in the sector No.14 for 3334.759200 ms

```

Далее протестируем с тем же количеством секторов (100), но с использованием 16 потоков. Результат так же верен.

```

Microsoft Visual Studio Debug Console

Input number of search sectors on the area (0 < x < 1000): 100
INFO: Money has been located in the sector No.14

Please input number of groups (0 < x < 16): 16
----- Silver's report-----
---> Pirates group 0 has got sector No.0
---> Pirates group 2 has got sector No.1
<<<< Pirates group 0 has NOT found money in the sector No.0, Spent time 0.005900 mcs for 41 size (0.000000)
---> Pirates group 0 has got sector No.2
---> Pirates group 3 has got sector No.3
---> Pirates group 1 has got sector No.4
---> Pirates group 5 has got sector No.5
---> Pirates group 4 has got sector No.6
---> Pirates group 7 has got sector No.7
---> Pirates group 6 has got sector No.8
---> Pirates group 8 has got sector No.9
---> Pirates group 9 has got sector No.10
---> Pirates group 10 has got sector No.11
---> Pirates group 13 has got sector No.12
---> Pirates group 12 has got sector No.13
---> Pirates group 11 has got sector No.14
---> Pirates group 14 has got sector No.15
---> Pirates group 15 has got sector No.16
<<<< Pirates group 14 has NOT found money in the sector No.15, Spent time 0.840600 mcs for 491 size (0.000000)
---> Pirates group 14 has got sector No.17
<<<< Pirates group 15 has NOT found money in the sector No.16, Spent time 31.325900 mcs for 2995 size (0.000000)
---> Pirates group 15 has got sector No.18
<<<< Pirates group 15 has NOT found money in the sector No.18, Spent time 81.436100 mcs for 4827 size (0.000000)
---> Pirates group 15 has got sector No.19
<<<< Pirates group 9 has NOT found money in the sector No.10, Spent time 116.906600 mcs for 5705 size (0.000000)
---> Pirates group 9 has got sector No.20
<<<< Pirates group 0 has NOT found money in the sector No.2, Spent time 140.171700 mcs for 6334 size (0.000000)
---> Pirates group 0 has got sector No.21
<<<< Pirates group 15 has NOT found money in the sector No.19, Spent time 103.784200 mcs for 5436 size (0.000000)
---> Pirates group 15 has got sector No.22
<<<< Pirates group 15 has NOT found money in the sector No.22, Spent time 53.223500 mcs for 3902 size (0.000000)
---> Pirates group 15 has got sector No.23
<<<< Pirates group 15 has NOT found money in the sector No.23, Spent time 0.081800 mcs for 153 size (0.000000)
---> Pirates group 15 has got sector No.24
<<<< Pirates group 15 has NOT found money in the sector No.24, Spent time 0.297400 mcs for 292 size (0.000000)
---> Pirates group 15 has got sector No.25
!!!! Pirates group 11 has found money in the sector No.14, Elapsed time 351.563900 mcs
<<<< Pirates group 4 has NOT found money in the sector No.6, Spent time 461.794000 mcs for 11478 size (0.000000)
<<<< Pirates group 14 has NOT found money in the sector No.17, Spent time 498.675600 mcs for 11942 size (0.000000)
<<<< Pirates group 15 has NOT found money in the sector No.25, Spent time 537.014900 mcs for 12382 size (0.000000)
<<<< Pirates group 5 has NOT found money in the sector No.5, Spent time 866.678800 mcs for 15724 size (0.000000)
<<<< Pirates group 0 has NOT found money in the sector No.21, Spent time 745.083800 mcs for 14604 size (0.000000)
<<<< Pirates group 12 has NOT found money in the sector No.13, Spent time 991.871300 mcs for 16827 size (0.000000)
<<<< Pirates group 2 has NOT found money in the sector No.1, Spent time 1192.365200 mcs for 18467 size (0.000000)
<<<< Pirates group 1 has NOT found money in the sector No.4, Spent time 1285.707300 mcs for 19169 size (0.000000)
<<<< Pirates group 13 has NOT found money in the sector No.12, Spent time 1896.143400 mcs for 23281 size (0.000000)
<<<< Pirates group 8 has NOT found money in the sector No.9, Spent time 2091.935200 mcs for 24464 size (0.000000)
<<<< Pirates group 3 has NOT found money in the sector No.3, Spent time 2456.018400 mcs for 26500 size (0.000000)
<<<< Pirates group 6 has NOT found money in the sector No.8, Spent time 2540.678400 mcs for 26962 size (0.000000)
<<<< Pirates group 10 has NOT found money in the sector No.11, Spent time 2765.346000 mcs for 28145 size (0.000000)
<<<< Pirates group 7 has NOT found money in the sector No.7, Spent time 3009.772600 mcs for 29358 size (0.000000)
<<<< Pirates group 9 has NOT found money in the sector No.20, Spent time 3654.639500 mcs for 32391 size (0.000000)

RESULTS: Pirates group 11 has found Money in the sector No.14 For 3774.752400 ms

```


Интересно, что для небольшого числа секторов при использовании 8 потоков происходит большее ускорение работы, чем при 16. Скорее всего это связано с тем, что время на инициализацию всех 16 потоков при таком небольшом кол-ве данных для обработки (14) больше, чем время выигрываемое за счёт распределения задач.

3.2.2. Тестирование на большем числе

Протестируем для числа больше. Кол-во секторов 400 и 4 поисковые группы. Клад находится в 216 секторе. Верно находится сектор с кладом. Также запустим программу с тестирования на 8 и 16 потоках.

```
Input number of search sectors on the area (0 < x < 1000): 400
INFO: Money has been located in the sector No.216

Please input number of groups (0 < x < 16): 4
----- Silver's report-----
---> Pirates group 1 has got sector No.0
---> Pirates group 0 has got sector No.1
---> Pirates group 2 has got sector No.2
<<<< Pirates group 1 has NOT found money in the sector No.0, Spent time 0.006000 mcs for 41 size (0.000000)
---> Pirates group 1 has got sector No.3
---> Pirates group 3 has got sector No.4

---> Pirates group 1 has got sector No.214
<<<< Pirates group 1 has NOT found money in the sector No.214, Spent time 784.530000 mcs for 14989 size (0.000000)
---> Pirates group 1 has got sector No.215
<<<< Pirates group 2 has NOT found money in the sector No.207, Spent time 3479.265400 mcs for 31556 size (0.000000)
---> Pirates group 2 has got sector No.216
!!!! Pirates group 2 has found money in the sector No.216, Elapsed time 35.717100 mcs
<<<< Pirates group 3 has NOT found money in the sector No.209, Spent time 3207.040300 mcs for 30303 size (0.000000)
<<<< Pirates group 0 has NOT found money in the sector No.213, Spent time 3716.362100 mcs for 32609 size (0.000000)
<<<< Pirates group 1 has NOT found money in the sector No.215, Spent time 3732.793800 mcs for 32702 size (0.000000)

RESULTS: Pirates group 2 has found Money in the sector No.216 for 68279.673500 ms
```

Результаты выполнения для 400 секторов на 8 потоках. Также интересно понаблюдать за занятостью 16 логических процессоров. При 8 потоках они используются на 51%.

```
Input number of search sectors on the area (0 < x < 1000): 400
INFO: Money has been located in the sector No.216

Please input number of groups (0 < x < 16): 8
----- Silver's report-----
---> Pirates group 0 has got sector No.0
---> Pirates group 1 has got sector No.1
<<<< Pirates group 0 has NOT found money in the sector No.0, Spent time 0.006400 mcs for 41 size (0.000000)
---> Pirates group 0 has got sector No.2
---> Pirates group 3 has got sector No.3
---> Pirates group 4 has got sector No.4
---> Pirates group 5 has got sector No.5
---> Pirates group 2 has got sector No.6
---> Pirates group 6 has got sector No.7
---> Pirates group 7 has got sector No.8
<<<< Pirates group 0 has NOT found money in the sector No.2, Spent time 140.040500 mcs for 6334 size (0.000000)
---> Pirates group 0 has got sector No.9
---> Pirates group 4 has got sector No.217
!!!! Pirates group 3 has found money in the sector No.216, Elapsed time 35.624800 mcs
<<<< Pirates group 2 has NOT found money in the sector No.214, Spent time 786.024100 mcs for 14989 size (0.000000)
<<<< Pirates group 6 has NOT found money in the sector No.208, Spent time 1824.656000 mcs for 22798 size (0.000000)
<<<< Pirates group 4 has NOT found money in the sector No.217, Spent time 1465.368000 mcs for 20485 size (0.000000)
<<<< Pirates group 5 has NOT found money in the sector No.209, Spent time 3212.167400 mcs for 30303 size (0.000000)
<<<< Pirates group 1 has NOT found money in the sector No.207, Spent time 3480.990500 mcs for 31556 size (0.000000)
<<<< Pirates group 7 has NOT found money in the sector No.213, Spent time 3713.601000 mcs for 32609 size (0.000000)
<<<< Pirates group 0 has NOT found money in the sector No.215, Spent time 3735.004000 mcs for 32702 size (0.000000)

RESULTS: Pirates group 3 has found Money in the sector No.216 for 35319.500900 ms
```

ЦП

Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz

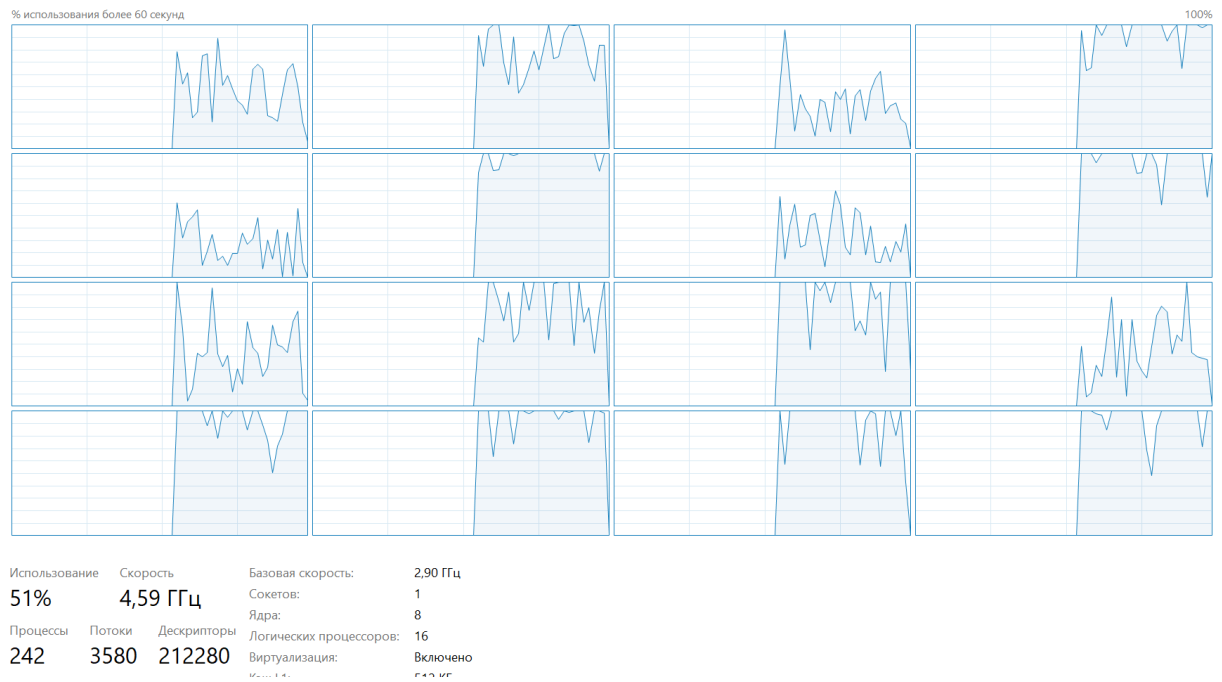


Рисунок 1 - Логические процессоры при использовании 8 потоков

Результаты выполнения для 400 секторов на 16 потоках.

```
Input number of search sectors on the area (0 < x < 1000): 400
INFO: Money has been located in the sector No.216

Please input number of groups (0 < x < 16): 16
----- Silver's report-----
----> Pirates group 0 has got sector No.0
----> Pirates group 1 has got sector No.1
<<<< Pirates group 0 has NOT found money in the sector No.0, Spent time 0.006000 mcs for 41 size (0.000000)
----> Pirates group 0 has got sector No.2
----> Pirates group 3 has got sector No.3
----> Pirates group 7 has got sector No.217
!!!! Pirates group 4 has found money in the sector No.216, Elapsed time 36.803800 mcs
<<<< Pirates group 2 has NOT found money in the sector No.193, Spent time 1797.625800 mcs for 22483 size (0.000000)
<<<< Pirates group 6 has NOT found money in the sector No.203, Spent time 1065.516100 mcs for 17451 size (0.000000)
<<<< Pirates group 11 has NOT found money in the sector No.206, Spent time 966.318500 mcs for 16519 size (0.000000)
<<<< Pirates group 5 has NOT found money in the sector No.192, Spent time 2175.457600 mcs for 24648 size (0.000000)
<<<< Pirates group 12 has NOT found money in the sector No.202, Spent time 1283.286700 mcs for 18935 size (0.000000)
<<<< Pirates group 3 has NOT found money in the sector No.185, Spent time 2817.411300 mcs for 27938 size (0.000000)
<<<< Pirates group 1 has NOT found money in the sector No.198, Spent time 1834.913500 mcs for 22813 size (0.000000)
<<<< Pirates group 10 has NOT found money in the sector No.214, Spent time 791.901500 mcs for 14989 size (0.000000)
<<<< Pirates group 0 has NOT found money in the sector No.204, Spent time 1517.506900 mcs for 20600 size (0.000000)
<<<< Pirates group 8 has NOT found money in the sector No.208, Spent time 1843.464100 mcs for 22798 size (0.000000)
<<<< Pirates group 7 has NOT found money in the sector No.217, Spent time 1469.588600 mcs for 20485 size (0.000000)
<<<< Pirates group 13 has NOT found money in the sector No.209, Spent time 3238.063500 mcs for 30303 size (0.000000)
<<<< Pirates group 9 has NOT found money in the sector No.207, Spent time 3484.804900 mcs for 31556 size (0.000000)
<<<< Pirates group 15 has NOT found money in the sector No.213, Spent time 3722.182200 mcs for 32609 size (0.000000)
<<<< Pirates group 14 has NOT found money in the sector No.215, Spent time 3735.433200 mcs for 32702 size (0.000000)

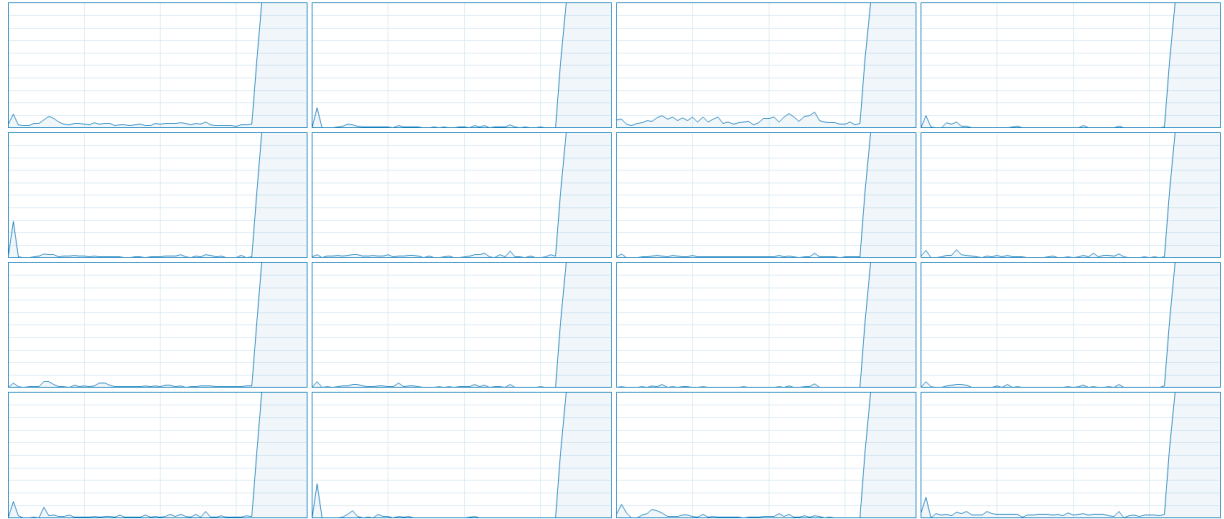
RESULTS: Pirates group 4 has found Money in the sector No.216 for 19649.776400 ms
```

При использовании 16 потоков, соответственно используются все 100%

ЦП

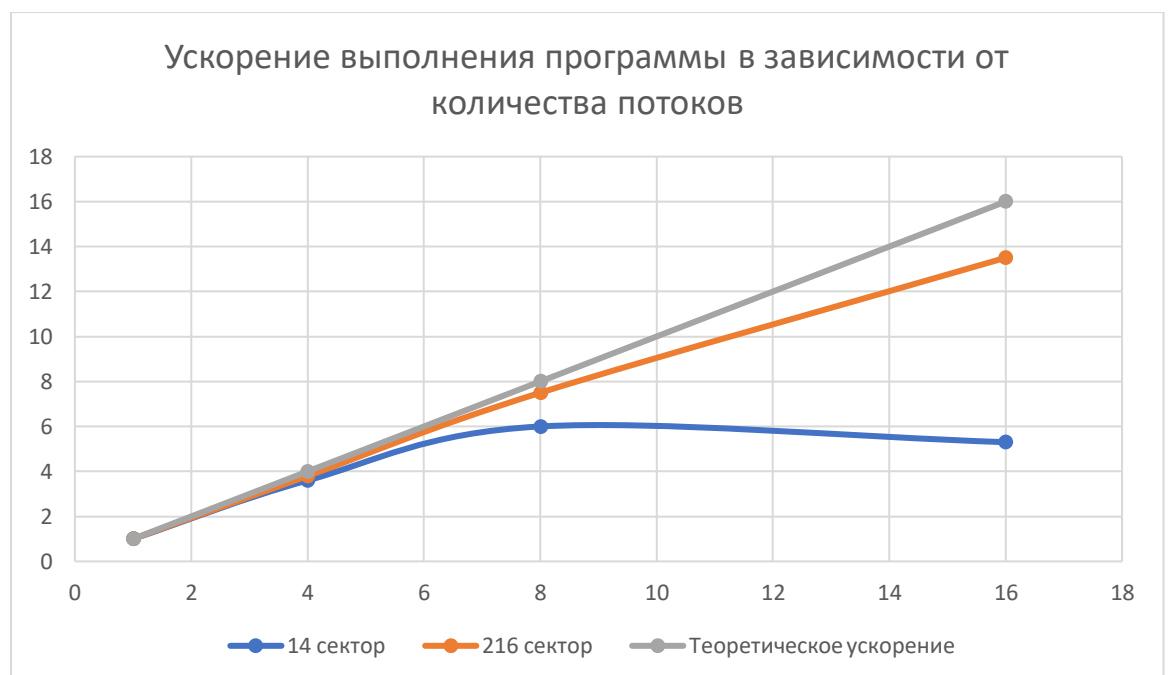
Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz

% использования более 60 секунд



| | | | |
|---------------|----------|-------------------|----------|
| Использование | Скорость | Базовая скорость: | 2,90 ГГц |
| 100% | 4,59 ГГц | Сокетов: | 1 |
| | | Ядра: | 8 |

На основе полученных при тестировании данных был составлен график зависимости ускорения выполнения программы в зависимости от количества потоков. Для небольших чисел (14) различия незначительны, в то время как для числа 216 выполнение в 16 потоков ускоряет вычисления в 13,5 раз, чем если бы выполнение происходило в 1 потоке.



4. ТЕКСТ ПРОГРАММЫ

```
5. #include <iostream>
6. #include <vector>
7. #include <thread>
8. #include <mutex>
9.
10. using namespace std;
11.
12. mutex mtx;
13.
14. int currentSector = 0;
15. int sectorsNum;
16. bool moneyFound = true;
17. int group = -1;
18.
19. struct Sector
20. {
21.     int size = 0;
22.     bool withMoney = false;
23. };
24.
25. Sector* sectors; // портфель задач
26.
27. // Время- и ресурсно- затратная функция
28. double uselessFunc(const int size)
29. {
30.     double d = 1;
31.     for (int i = 0; i < size; ++i) {
32.         for (int j = 0; j < size; ++j)
33.             d += d * i * j - d;
34.     }
35.     return d;
36. }
37.
38. //Запрос числа секторов от пользователя
39. int getNoOfSectors()
40. {
41.     cout << "Input number of search sectors on the area (0 < x < 1000): ";
42.     int num;
43.     cin >> num;
44.     while (num <= 0 or num > 1000)
45.     {
46.         printf("Incorrect input! Try again: ");
47.         cin >> num;
48.     }
49.     return num;
50. }
51.
52. //Запрос числа групп от пользователя
53. int getNoOfGroups()
54. {
55.     // Кол-во одновременных потоков, поддерживаемых системой
56.     unsigned int n = std::thread::hardware_concurrency();
57.
58.     printf("Please input number of groups (0 < x <= %d): ", n);
59.
60.     int num;
61.     cin >> num;
62.     while (num <= 0 or num > n)
63.     {
64.         printf("Incorrect input! Try again: ");
65.         cin >> num;
66.     }
67.     return num;
68. }
```

```

69.
70. void* findMoney(int name)
71. {
72.     //ищем пока клад не найден
73.     while (moneyFound)
74.     {
75.         //Сильвер берет сектор из списка секторов и выдает задание группе Name
76.         //блокируем для других групп
77.         mtx.lock();
78.         //Сильвер дает сектор i текущей группе и увеличивает счетчик
79.         int i = currentSector++;
80.         printf("---> Pirates group %d has got sector No.%d \n", name, i);
81.         mtx.unlock();
82.         //Получаем значение данного сектора в массиве (есть там клад или нет): 1 -
            присутствует 0 - отсутствует
83.
84.         //Проверка секторов - занимает разное время
85.         auto t_start = std::chrono::high_resolution_clock::now();
86.         double d = uselessFunc(sectors[i].size); // нагружаем группу объемом
            работы, зависящим от размера поискового сектора
87.         auto t_end = std::chrono::high_resolution_clock::now();
88.         double dt = std::chrono::duration<double, std::milli>(t_end -
            t_start).count();
89.
90.         if (sectors[i].withMoney)
91.         {
92.             // блокируем поток на вывод в консоль
93.             mtx.lock();
94.             printf("!!!! Pirates group %d has found money in the sector No.%d,
                Elapsed time %f mcs \n", name, i, dt);
95.             mtx.unlock();
96.             moneyFound = false;
97.             group = name;
98.         }
99.         else
100.        {
101.            mtx.lock();
102.            printf("<<<< Pirates group %d has NOT found money in the sector
                No.%d, Spent time %f mcs for %d size (%f)\n", name, i, dt, sectors[i].size, d);
103.            mtx.unlock();
104.        }
105.
106.    }
107.    return nullptr;
108. }
109.
110.
111. int main()
112. {
113.     //получаем количество секторов
114.     sectorsNum = getNoOfSectors();
115.
116.     //инициализируем секторы острова
117.     sectors = new Sector[sectorsNum];
118.
119.     // генерируем сектора острова - у каждого свой размер.
120.     for (int i = 0; i < sectorsNum; i++)
121.     {
122.         sectors[i] = Sector{ rand(), false };
123.     }
124.
125.     // рандомно выбираем сектор с кладом
126.     int sectorId = (double)rand() / RAND_MAX * sectorsNum;
127.     sectors[sectorId].withMoney = true;
128.

```

```

129.         // выводим информацию о номере выбранного сектора с кладом
130.         printf("INFO: Money has been located in the sector No.%d \n\n",
sectorId);
131.
132.         // поисковые группы пиратов симулируются количеством созданных потоков
133.         int threadNumber = getNoOfGroups();
134.         vector<thread> thr(threadNumber);
135.
136.         // создание потоков
137.         printf("----- Silver's report-----\n");
138.         auto t_start = std::chrono::high_resolution_clock::now();
139.         for (int i = 0; i < threadNumber; i++)
140.         {
141.             thr[i] = thread(findMoney, i);
142.         }
143.
144.         for (int i = 0; i < threadNumber; i++)
145.         {
146.             thr[i].join();
147.         }
148.
149.         auto t_end = std::chrono::high_resolution_clock::now();
150.         printf("\nRESULTS: Pirates group %d has found Money in the sector No.%d
for %f ms \n", group, sectorId,
151.             std::chrono::duration<double, std::milli>(t_end -
t_start).count());
152.
153.         return 0;
154.     }
155.

```

5. СПИСОК ИСТОЧНИКОВ

- 1) Модели взаимодействия процессов. Управляющий – рабочие (распределенный портфель задач) [Электронный ресурс] // Режим доступа: <https://l.wzm.me/coder/custom/parallel.programming/009.htm> , свободный (Дата обращения: 16.11.2020)
- 2) Потоки, блокировки и условные переменные в C++ [Электронный ресурс] // Режим доступа: <https://habr.com/ru/post/182610/> , свободный (Дата обращения: 16.11.2020)
- 3) Парадигмы параллельного программирования [Электронный ресурс] // Режим доступа: <https://pro-prof.com/forums/topic/parallel-programming-paradigms>, свободный (Дата обращения: 16.11.2020)