Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №2 по курсу

«Операционные системы»

Группа: М8О-211Б-23

Студент: Тремель Д.А.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 27.11.24

Москва, 2024

Постановка задачи

Вариант 19.

Цель работы

Целью является приобретение практических навыков в:

- Управление потоками в ОС
- Обеспечение синхронизации между потоками

Задание

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение максимального количества потоков, работающих в один момент времени, должно быть задано ключом запуска вашей программы. Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы. В отчете привести исследование зависимости ускорения и эффективности алгоритма от входных данных и количества потоков. Получившиеся результаты необходимо объяснить.

Дан массив координат (x, y). Пользователь вводит число кластеров. Проведите кластеризацию методом k-средних

Общий метод и алгоритм решения

Использованные системные вызовы:

- ssize_t write(int __fd, const void *__buf, size_t __n); — Записывает N байт из буфера (BUF) в файл (FD). Возвращает количество записанных байт или -1.
- void exit(int __status);— Выполняет немедленное завершение программы. Все используемые программой потоки закрываются, и временные файлы удаляются, управление возвращается ОС или другой программе.
- int pthread_create(pthread_t *__restrict__ __newthread, const pthread_attr_t *__restrict__ __attr, void *(*__start_routine)(void *), void *__restrict__ __arg); — Создаёт поток с рутиной (стартовой функцией) и заданными аргументами.
- int pthread_join(pthread_t __th, void **__thread_return); — Дожидается завершения потока.

Для реализации мьютексов были использованы:

- pthread_mutex_t — тип данных для мьютекса.
- int pthread_mutex_init(pthread_mutex_t *mutex, const pthread_mutexattr_t *mutexattr); —Инициализация мьютекса.
- int pthread_mutex_lock(pthread_mutex_t *mutex); — Блокировка мьютекса.
- int pthread_mutex_unlock(pthread_mutex_t *mutex); — Разблокировка мьютекса.
- int pthread_mutex_destroy(pthread_mutex_t *mutex);— Удаление мьютекса.

Решение реализует алгоритм кластеризации k-средних с использованием многопоточности для повышения производительности. Программа считывает координаты точек из файла, затем инициализирует центроиды кластеров на основе первых точек. На каждой итерации точки распределяются по кластерам в потоках, где синхронизация обновления данных обеспечивается с помощью мьютексов. После распределения центроиды обновляются на основе средних значений координат точек в каждом кластере. Итерации продолжаются до тех пор, пока центроиды не стабилизируются или не будет достигнут лимит итераций.

Код программы

main.c

```
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <pthread.h>
#include <errno.h>
#include <stdio.h>
#include <time.h>

#define MAX_POINTS 10000
#define MAX_CLUSTERS 100

typedef struct {
    double x, y;
} Point;

typedef struct {
    int start, end;
    Point* points;
    Point* centroids;
    int* point_cluster;
    int num_clusters;
} Thread_data;



pthread_mutex_t mutexes[MAX_CLUSTERS];
double cluster_sums_x[MAX_CLUSTERS];
double cluster_sums_y[MAX_CLUSTERS];
int cluster_counts[MAX_CLUSTERS];

void write_message(const char* message) {
    write(STDOUT_FILENO, message, strlen(message));
}

void* assign_clusters(void* arg) {
    Thread_data* data = (Thread_data*)arg;

    for (int i = data->start; i < data->end; i++) {
```

```
        double min_dist = INFINITY;
        int closest_cluster = -1;

        for (int j = 0; j < data->num_clusters; j++)
        {
            double dist = sqrt(pow(data->points[i].x - data->centroids[j].x, 2) +
                               pow(data->points[i].y - data->centroids[j].y, 2));
            if (dist < min_dist) {
                min_dist = dist;
                closest_cluster = j;
            }
        }

        data->point_cluster[i] = closest_cluster;

        pthread_mutex_lock(&mutexes[closest_cluster]);
        cluster_sums_x[closest_cluster] += data->points[i].x;
        cluster_sums_y[closest_cluster] += data->points[i].y;
        cluster_counts[closest_cluster]++;
        pthread_mutex_unlock(&mutexes[closest_cluster]);
    }

    return NULL;
}

void update_centroids(Point* centroids, int num_clusters) {
    for (int i = 0; i < num_clusters; i++) {
        if (cluster_counts[i] > 0) {
            centroids[i].x = cluster_sums_x[i] / cluster_counts[i];
            centroids[i].y = cluster_sums_y[i] / cluster_counts[i];
        }
    }
}

int main(int argc, char* argv[]) {
    if (argc < 3) {
        write_message("Usage: ./program <num_clusters> <num_threads>\n");
        return 1;
    }

    int num_clusters = atoi(argv[1]);
    int num_threads = atoi(argv[2]);

    if (num_clusters > MAX_CLUSTERS || num_threads < 1) {
        write_message("Invalid number of clusters or threads.\n");
        return 1;
    }

    int num_points = 1000;
```

```
    Point points[num_points];
    Point centroids[num_clusters];
    Point prev_centroids[num_clusters];
    int point_cluster[num_points];

    pthread_t threads[num_threads];
    Thread_data thread_data[num_threads];

    FILE* file = fopen("test", "r");

    for (int i = 0; i < num_points; i++) {
        if (fscanf(file, "%lf %lf", &points[i].x, &points[i].y) != 2) {
            fprintf(stderr, "Error reading point %d from file\n", i);
            fclose(file);
            return 1;
        }
    }

    fclose(file);

    for (int i = 0; i < num_clusters; i++) {
        pthread_mutex_init(&mutexes[i], NULL);
    }

    clock_t start_time = clock();

    for (int i = 0; i < num_clusters; i++) {
        centroids[i] = points[i];
    }

    int flag = 0, iterations = 0;
    while (!flag) {
        for (int i = 0; i < num_clusters; i++) {
            prev_centroids[i] = centroids[i];
        }

        memset(cluster_sums_x, 0, sizeof(cluster_sums_x));
        memset(cluster_sums_y, 0, sizeof(cluster_sums_y));
        memset(cluster_counts, 0, sizeof(cluster_counts));


        int chunk_size = (num_points + num_threads - 1) / num_threads;
        for (int i = 0; i < num_threads; i++) {
            thread_data[i].start = i * chunk_size;
            thread_data[i].end = (i + 1) * chunk_size > num_points ? num_points
: (i + 1) * chunk_size;
            thread_data[i].points = points;
            thread_data[i].centroids = centroids;
            thread_data[i].point_cluster = point_cluster;
```

```
            thread_data[i].num_clusters = num_clusters;

            pthread_create(&threads[i], NULL, assign_clusters,
 &thread_data[i]);
        }

        for (int i = 0; i < num_threads; i++) {
            pthread_join(threads[i], NULL);
        }

        update_centroids(centroids, num_clusters);

        flag = 1;
        for (int i = 0; i < num_clusters; i++) {
            if (fabs(centroids[i].x - prev_centroids[i].x) > 1e-4 ||
                fabs(centroids[i].y - prev_centroids[i].y) > 1e-4) {
                flag = 0;
                break;
            }
        }

        iterations++;

        if (iterations > 1000) {
            write_message("Iteration limit reached. Stopping.\n");
            break;
        }
    }

    printf("Clustering completed in %d iterations.\n", iterations);
    for (int j = 0; j < num_clusters; ++j)
    {
        printf("Cluster %d: %lf %lf\n", j + 1, centroids[j].x, centroids[j].y);
    }

    printf("Execution time: %.6f seconds\n", (double)(clock() - start_time)
 / CLOCKS_PER_SEC);

    for (int i = 0; i < num_clusters; i++) {
        pthread_mutex_destroy(&mutexes[i]);
    }


    return 0;
}
```

Протокол работы программы

```
u@DESKTOP-3U3OER0:/mnt/c/Users/u/CLionProjects/OS/lab_2$ ./a 5 1
Clustering completed in 18 iterations.
Cluster 1: 74.713469 25.993469
Cluster 2: 83.279618 74.971338
Cluster 3: 15.118293 74.473171
Cluster 4: 22.001224 24.732653
Cluster 5: 47.330159 71.500529
Execution time: 0.012387 seconds
u@DESKTOP-3U3OER0:/mnt/c/Users/u/CLionProjects/OS/lab_2$ ./a 5 3
Clustering completed in 18 iterations.
Cluster 1: 74.713469 25.993469
Cluster 2: 83.279618 74.971338
Cluster 3: 15.118293 74.473171
Cluster 4: 22.001224 24.732653
Cluster 5: 47.330159 71.500529
Execution time: 0.080191 seconds
u@DESKTOP-3U3OER0:/mnt/c/Users/u/CLionProjects/OS/lab_2$ ./a 5 5
Clustering completed in 18 iterations.
Cluster 1: 74.713469 25.993469
Cluster 2: 83.279618 74.971338
Cluster 3: 15.118293 74.473171
Cluster 4: 22.001224 24.732653
Cluster 5: 47.330159 71.500529
Execution time: 0.107540 seconds
u@DESKTOP-3U3OER0:/mnt/c/Users/u/CLionProjects/OS/lab_2$ ./a 5 8
Clustering completed in 18 iterations.
Cluster 1: 74.713469 25.993469
Cluster 2: 83.279618 74.971338
Cluster 3: 15.118293 74.473171
Cluster 4: 22.001224 24.732653
```

```
Cluster 5: 47.330159 71.500529
Execution time: 0.188274 seconds
u@DESKTOP-3U3OER0:/mnt/c/Users/u/CLionProjects/OS/lab_2$ ./a 5 10
Clustering completed in 18 iterations.
Cluster 1: 74.713469 25.993469
Cluster 2: 83.279618 74.971338
Cluster 3: 15.118293 74.473171
Cluster 4: 22.001224 24.732653
Cluster 5: 47.330159 71.500529
Execution time: 0.215350 seconds
u@DESKTOP-3U3OER0:/mnt/c/Users/u/CLionProjects/OS/lab_2$ ./a 5 12
Clustering completed in 18 iterations.
Cluster 1: 74.713469 25.993469
Cluster 2: 83.279618 74.971338
Cluster 3: 15.118293 74.473171
Cluster 4: 22.001224 24.732653
Cluster 5: 47.330159 71.500529
Execution time: 0.255253 seconds
u@DESKTOP-3U3OER0:/mnt/c/Users/u/CLionProjects/OS/lab_2$
```

Strace

```
u@DESKTOP-3U3OER0:/mnt/c/Users/u/CLionProjects/OS/lab_2$ strace ./a 3 3
execve("./a", ["./a", "3", "3"], 0x7ffe0d847580 /* 26 vars */) = 0
brk(NULL)                               = 0x55c63c441000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffdb0f132b0) = -1 EINVAL (Invalid
argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=57777, ...}) = 0
mmap(NULL, 57777, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f7faa23a000
close(3)                                = 0
```

```
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC)
= 3

read(3,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300\323\0\0\0\0\0\0"...,
832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=1369384, ...}) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x7f7faa238000

mmap(NULL, 1368336, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f7faa0e9000

mmap(0x7f7faa0f6000, 684032, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xd0

00) = 0x7f7faa0f6000

mmap(0x7f7faa19d000, 626688, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xb4000) = 0x7

f7faa19d000

mmap(0x7f7faa236000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x14c

000) = 0x7f7faa236000

close(3)                                = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0",
O_RDONLY|O_CLOEXEC) = 3

read(3,
"\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220q\0\0\0\0\0\0"...,
832) = 832

pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\232e\273F\236E\241\306\373\317\372\345\2
70*/\327"..

., 68, 824) = 68

fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0

pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\232e\273F\236E\241\306\373\317\372\345\2
70*/\327"..

., 68, 824) = 68

mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f7faa0c6000

mmap(0x7f7faa0cc000, 69632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x600

0) = 0x7f7faa0cc000
```

```
mmap(0x7f7faa0dd000, 24576, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17000) = 0x7f

7faa0dd000

mmap(0x7f7faa0e3000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c0

00) = 0x7f7faa0e3000

mmap(0x7f7faa0e5000, 13432, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0)

= 0x7f7faa0e5000

close(3)                                = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC)
= 3

read(3,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0"...,
832) = 832

pread64(3,
"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784

pread64(3,
"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32,
848) = 32

pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357_\243\335\2449\206V>\237\374\3
04"..., 68,

 880) = 68

fstat(3, {st_mode=S_IFREG|0755, st_size=2029592, ...}) = 0

pread64(3,
"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784

pread64(3,
"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32,
848) = 32

pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357_\243\335\2449\206V>\237\374\3
04"..., 68,

 880) = 68

mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f7fa9ed4000

mmap(0x7f7fa9ef6000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2
```

```
2000) = 0x7f7fa9ef6000

mmap(0x7f7faa06e000, 319488, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19a000) = 0x

7f7faa06e000

mmap(0x7f7faa0bc000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e

7000) = 0x7f7faa0bc000

mmap(0x7f7faa0c2000, 13920, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0)

= 0x7f7faa0c2000

close(3)                                = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7f7fa9ed1000

arch_prctl(ARCH_SET_FS, 0x7f7fa9ed1740) = 0

mprotect(0x7f7faa0bc000, 16384, PROT_READ) = 0

mprotect(0x7f7faa0e3000, 4096, PROT_READ) = 0

mprotect(0x7f7faa236000, 4096, PROT_READ) = 0

mprotect(0x55c63a886000, 4096, PROT_READ) = 0

mprotect(0x7f7faa276000, 4096, PROT_READ) = 0

munmap(0x7f7faa23a000, 57777)           = 0

set_tid_address(0x7f7fa9ed1a10)         = 6998

set_robust_list(0x7f7fa9ed1a20, 24)     = 0

rt_sigaction(SIGRTMIN, {sa_handler=0x7f7faa0ccbf0, sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO

, sa_restorer=0x7f7faa0da420}, NULL, 8) = 0

rt_sigaction(SIGRT_1, {sa_handler=0x7f7faa0ccc90, sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|

SA_SIGINFO, sa_restorer=0x7f7faa0da420}, NULL, 8) = 0

rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0

brk(NULL)                               = 0x55c63c441000

brk(0x55c63c462000)                     = 0x55c63c462000

openat(AT_FDCWD, "test", O_RDONLY)      = 3

fstat(3, {st_mode=S_IFREG|0777, st_size=332832, ...}) = 0
```

```
read(3, "89.700000 80.200000\r\n76.500000 9"..., 512) = 512

read(3, "74.300000\r\n97.600000 90.800000\r\n"..., 512) = 512

read(3, "21.400000 19.600000\r\n69.000000 5"..., 512) = 512

read(3, "700000\r\n15.700000 30.400000\r\n48."..., 512) = 512

read(3, "000 76.600000\r\n46.600000 67.3000"..., 512) = 512

read(3, "0\r\n32.000000 75.500000\r\n17.60000"..., 512) = 512

read(3, "000000\r\n82.300000 78.700000\r\n42."..., 512) = 512

...

close(3)                                = 0

clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0, tv_nsec=28081000}) =
0

mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0) = 0x7f7fa96d0000

mprotect(0x7f7fa96d1000, 8388608, PROT_READ|PROT_WRITE) = 0

clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 6999

mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0) = 0x7f7fa8ecf000

mprotect(0x7f7fa8ed0000, 8388608, PROT_READ|PROT_WRITE) = 0

clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7000

mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0) = 0x7f7fa86ce000

mprotect(0x7f7fa86cf000, 8388608, PROT_READ|PROT_WRITE) = 0

clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7001

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7001, NULL) = 0
```

```
clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7002

clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7003

clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7004

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7003, NULL) = 0

futex(0x7f7fa9ed09d0, FUTEX_WAIT, 7004, NULL) = 0

clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7005

clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7006

clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7007

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7006, NULL) = 0

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7007, NULL) = 0

clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE
```

**AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8**

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7008

**clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE**

**AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9**

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7009

**clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE**

**AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9**

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7010

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7008, NULL) = 0

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7009, NULL) = 0

futex(0x7f7fa9ed09d0, FUTEX_WAIT, 7010, NULL) = 0

**clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE**

**AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9**

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7011

**clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE**

**AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9**

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7012

**clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE**

**AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8**

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7013

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7012, NULL) = 0

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7013, NULL) = 0

**clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE**

**AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8**

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7014

clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7015

clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7016

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7014, NULL) = -1 EAGAIN (Resource temporarily unavailable)

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7015, NULL) = -1 EAGAIN (Resource temporarily unavailable)

futex(0x7f7fa9ed09d0, FUTEX_WAIT, 7016, NULL) = -1 EAGAIN (Resource temporarily unavailable)

clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7017

clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7018

clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7019

futex(0x7f7fa9ed09d0, FUTEX_WAIT, 7017, NULL) = 0

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7018, NULL) = 0

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7019, NULL) = -1 EAGAIN (Resource temporarily unavailable)

clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7020

clone(child_stack=0x7f7fa96cefb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7021

clone(child_stack=0x7f7fa9ecffb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7022

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7020, NULL) = -1 EAGAIN (Resource temporarily unavailable)

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7021, NULL) = 0

clone(child_stack=0x7f7fa9ecffb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7023

clone(child_stack=0x7f7fa96cefb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7024

clone(child_stack=0x7f7fa8ecdfb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7025

futex(0x7f7fa9ed09d0, FUTEX_WAIT, 7023, NULL) = 0

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7024, NULL) = -1 EAGAIN (Resource temporarily unavailable)

clone(child_stack=0x7f7fa8ecdfb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7026

clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7027

clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7028

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7026, NULL) = 0

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7027, NULL) = 0

futex(0x7f7fa9ed09d0, FUTEX_WAIT, 7028, NULL) = -1 EAGAIN (Resource temporarily unavailable)

clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7029

clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7030

clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7031

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7030, NULL) = 0

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7031, NULL) = -1 EAGAIN (Resource temporarily unavailable)

clone(child_stack=0x7f7fa8ecdfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa8

ece9d0, tls=0x7f7fa8ece700, child_tidptr=0x7f7fa8ece9d0) = 7032

clone(child_stack=0x7f7fa96cefb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

6cf9d0, tls=0x7f7fa96cf700, child_tidptr=0x7f7fa96cf9d0) = 7033

clone(child_stack=0x7f7fa9ecffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE

AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tidptr=0x7f7fa9

ed09d0, tls=0x7f7fa9ed0700, child_tidptr=0x7f7fa9ed09d0) = 7034

futex(0x7f7fa8ece9d0, FUTEX_WAIT, 7032, NULL) = -1 EAGAIN (Resource temporarily unavailable)

futex(0x7f7fa96cf9d0, FUTEX_WAIT, 7033, NULL) = 0

futex(0x7f7fa9ed09d0, FUTEX_WAIT, 7034, NULL) = -1 EAGAIN (Resource temporarily unavailable)

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0

write(1, "Clustering completed in 12 itera"..., 39Clustering completed in 12 iterations.

) = 39

write(1, "Cluster 1: 76.271935 32.829677\n", 31Cluster 1: 76.271935 32.829677

) = 31

write(1, "Cluster 2: 49.092541 79.770442\n", 31Cluster 2: 49.092541 79.770442

) = 31

write(1, "Cluster 3: 20.667683 33.144817\n", 31Cluster 3: 20.667683 33.144817

) = 31

clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0, tv_nsec=48100400}) = 0

write(1, "Execution time: 0.020019 seconds"..., 33Execution time: 0.020019 seconds

) = 33

exit_group(0)   = ?

+++ exited with 0 +++

Число потоков	Время исполнения (с)	Ускорение	Эффективность
1	0.012387	1	1
3	0.080191	0.15	0.05
5	0.107540	0.12	0.02
8	0.188274	0.07	0.01
10	0.215350	0.06	0.01
12	0.255253	0.05	0.004

Вывод

Программа выполняет кластеризацию методом k-средних с использованием многопоточности для распределения вычислений. Для синхронизации потоков применяются мьютексы, что обеспечивает корректное обновление данных кластеров. С увеличением количества потоков увеличивается и время выполнения, так как мы тратим время на ожидание завершения работы каждого потока, а также на закрытие каждого потока.