

Государственное образовательное учреждение высшего профессионального образования

«Московский Государственный Технический Университет имени Н. Э. Баумана»

ОТЧЕТ

По лабораторной работе №5

По курсу «Анализ алгоритмов»

Тема: **«Конвеерная обработка данных»**

Студент: Кононенко С. Д.

Группа: ИУ7-51

Москва, 2017

Постановка задачи

1. Реализовать конвеерную обработку данных с помощью потоков

2. Сравнить реализованный конвеер с однопоточной обработкой тех же данных

Теоретическая оценка

Пусть существует N вещественных чисел, каждая операция над числом занимает t времени, тогда

1. F(N) = N \* t \* 3 = 3tN, для однопоточной обработки

2. F(N) = (N + 2) \* t = tN + t, для конвеерной обработки

Реализация

Для реализации конвеерной обработки с помощью потоков было выделено три этапа, на каждом из которых производится своя операция обработки данных.

В качестве входных данных выбраны вещественные числа, в качетсве операций – функции pow, sqrt и sin.

Данные хранятся в очереди, каждый поток имеет доступ к 2-м очередям: из первой он берет данные на обработку, а во вторую передаёт обработаные данные.

Однопоточная обработка данных

queue\_arr process\_data(queue\_arr \*q, unsigned long long \*t)

{

\*t = tick();

queue\_arr res\_q = create\_queue\_arr(q->size);

while (q->count)

{

float num = pop\_q\_arr(q);

f1(&num);

f2(&num);

f3(&num);

push\_q\_arr(&res\_q, num);

}

\*t = tick() - \*t;

return res\_q;

}

Конвеер

static void \*conv1(void \*args)

{

convargs\_t arg = \*(convargs\_t \*)args;

while (arg.todo->count || !\*arg.pf)

{

while (arg.todo->count)

{

float num = pop\_q\_arr(arg.todo);

f1(&num);

EnterCriticalSection(&cs1);

push\_q\_arr(arg.next, num);

LeaveCriticalSection(&cs1);

}

}

\*arg.nf = 1;

}

*// conv2 и conv3 аналогичны conv1*

queue\_arr process\_data\_par(queue\_arr \*q, unsigned long long \*t)

{

InitializeCriticalSection(&cs1);

InitializeCriticalSection(&cs2);

\*t = tick();

pthread\_t t1, t2, t3;

queue\_arr q2 = create\_queue\_arr(q->size),

q3 = create\_queue\_arr(q->size),

res\_q = create\_queue\_arr(q->size);

int f0 = 1, f1 = 0, f2 = 0, f3 = 0;

convargs\_t args1 = {q, &q2, &f0, &f1}, args2 = {&q2, &q3, &f1, &f2},

args3 = {&q3, &res\_q, &f2, &f3};

;

pthread\_create(&t1, NULL, conv1, &args1);

pthread\_create(&t2, NULL, conv2, &args2);

pthread\_create(&t3, NULL, conv3, &args3);

pthread\_join(t1, NULL);

pthread\_join(t2, NULL);

pthread\_join(t3, NULL);

\*t = tick() - \*t;

return res\_q;

}

Эксперимент

*По оси ординат используется время работы в тиках \*1e-6*

*По оси абсцисс – количество эллементов в очереди*

Рисунок 1 Гистограмма сравнения эффективности однопоточной и конвеерной обработки

**Вывод:** к сожалению в данном примере из-за недостаточно высокой вычислительной сложности операций затраты на обслуживание конвеера практически скомпенсировали эффект от алгоритма, но всё равно можно сделать вывод об эффективности конвеера перед однопоточной обработкой

Заключение

В ходе лабораторной работы был реализован конвеерную обработку данных при помощи потоков. Также было проведено сравнение эффективности конвеерной и однопоточной обработки.