**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

**Факультет программной инженерии и компьютерной техники**

Отчет по лабораторной работе №1

по дисциплине «Параллельное программирование»

***Тема* «Автоматическое распараллеливание программ»**

Выполнили:

студенты гр. № P33212

Зиганшин Геннадий Муратович

Ян Цзяфэн

Проверил:

Балакшин Павел Валерьевич

Санкт-Петербург

2021

Содержание

[Варианты заданий 3](#_Toc66736344)

[Описание задачи 3](#_Toc66736345)

[Краткая характеристика 4](#_Toc66736346)

[Коды программы 4](#_Toc66736347)

[Таблица и графики значений функций seq(N), par-K(N) 4](#_Toc66736348)

[Вывод 6](#_Toc66736349)

# Варианты заданий

A = 576

Операция для массива M1 (вариант 6): Кубический корень после деления на число e

Операция для массива M2 (вариант 5): Натуральный логарифм модуля тангенса

Операция в этапе Merge (вариант 1): Возведение в степень (т.е. M2[i] = M1[i]^M2[i])

Операция в этапе Sort (вариант 6): Сортировка вставками (Insertion sort)

# Описание задачи

1. На компьютере с многоядерным процессором установить ОС Linux и компилятор GCC версии не ниже 4.7.2.
2. На языке Cи написать консольную программу lab1.c, решающую задачу ниже:

**Этап Generate.** Сформировать массив М1 размерностью N, заполнив его с помощью функции rand\_r случайными вещественными числами, имеющими равномерный закон распределения в диапазоне от 1 до A (включительно). Аналогично сформировать массив М2 размерностью N/2 со случайными вещественными числами в диапазоне от А до 10\*А.

**Этап Map.** В массиве М1 к каждому элементу применить операцию: кубический корень после деления на число e. Затем в массиве М2 каждый элемент поочерёдно сложить с предыдущим (для этого вам понадобится копия массива М2, из которого нужно будет брать операнды), а к результату сложения применить операцию (считать, что для начального элемента массива предыдущий элемент равен нулю): Натуральный логарифм модуля тангенса.

**Этап Merge.** В массивах М1 и М2 ко всем элементами с одинаковыми индексами попарно применить операцию (результат записать в М2): Возведение в степень (т.е. M2[i] = M1[i]^M2[i]).

**Этап Sort.** Полученный массив необходимо отсортировать методом: Сортировка вставками (Insertion sort).

**Этап Reduce.** Рассчитать сумму синусов тех элементов массива М2, которые при делении на минимальный ненулевой элемент массива М2 дают чётное число (при определении чётности учитывать только целую часть числа). Результатом работы программы по окончании пятого этапа должно стать одно число X, которое следует использовать для верификации программы после внесения в неё изменений (например, до и после распараллеливания итоговое число X не должно измениться в пределах погрешности). Значение числа X следует привести в отчёте для различных значений N.

1. Скомпилировать написанную программу без использования авто75 матического распараллеливания с помощью следующей команды: /home/user/gcc -O3 -Wall -Werror -o lab1-seq lab1.c
2. Скомпилировать написанную программу, используя встроенное в gcc средство автоматического распараллеливания Graphite с помощью следующей команды “/home/user/gcc -O3 -Wall -Werror -floopparallelize-all -ftree-parallelize-loops=K lab1.c -o lab1-par-K” (переменной K поочерёдно присвоить хотя бы 4 различных целых значений, выбор обосновать).
3. Запускать файл lab1-seq из командной строки, увеличивая значения N до значения N1, при котором время выполнения превысит 0.01 с. Подобным образом найти значение N=N2, при котором время выполнения превысит 2 с.
4. Используя найденные значения N1 и N2, выполнить следующие эксперименты (для автоматизации проведения экспериментов рекомендуется написать скрипт):

* запускать lab1-seq для значений N = N1, N1 + ∆, N1 + 2∆, N1 + 3∆, …, N2 и записывать получающиеся значения времени delta\_ms(N) в функцию seq(N);
* запускать lab1-par-K для значений N = N1, N1 + ∆, N1 + 2∆, N1 + 3∆, ..., N2 и записывать получающиеся значения времени delta\_ms(N) в функцию par − K(N);
* значение ∆ выбрать так: ∆ = (N2 − N1)/10.

# Краткая характеристика

* Процессор: AMD Ryzen 5 3600 6-Core Processor
* Операционная система: 20.04.1-Ubuntu
* Компилятор GCC: gcc 9.3.0
* ОЗУ 32 гб

# Коды программы

lab1.c в виде отдельного файла.

# Таблица и графики значений функций seq(N), par-K(N)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **Milliseconds passsed** | | | | | **S(K)** | | | | **Sum** |
| **seq(N)** | **par-2(N)** | **par-4(N)** | **par-8(N)** | **par-16(N)** | **S(2)** | **S(4)** | **S(8)** | **S(16)** |
| 880 | 14 | 12 | 8 | 10 | 6 | 1,167 | 1,750 | 1,400 | 2,333 | 0.366433 |
| 2792 | 56 | 59 | 55 | 56 | 82 | 0,949 | 1,018 | 1,000 | 0,683 | -20.339242 |
| 4704 | 148 | 148 | 153 | 153 | 210 | 1,000 | 0,967 | 0,967 | 0,705 | -21.621249 |
| 6616 | 284 | 285 | 297 | 295 | 349 | 0,996 | 0,956 | 0,963 | 0,814 | 31.322422 |
| 8528 | 467 | 464 | 477 | 491 | 549 | 1,006 | 0,979 | 0,951 | 0,851 | -64.746301 |
| 10440 | 683 | 681 | 693 | 690 | 758 | 1,003 | 0,986 | 0,990 | 0,901 | -41.852062 |
| 12352 | 956 | 952 | 967 | 960 | 1009 | 1,004 | 0,989 | 0,996 | 0,947 | -34.972800 |
| 14264 | 1268 | 1261 | 1282 | 1283 | 1324 | 1,006 | 0,989 | 0,988 | 0,958 | -30.783536 |
| 16176 | 1614 | 1634 | 1632 | 1639 | 1648 | 0,988 | 0,989 | 0,985 | 0,979 | -42.114233 |
| 18088 | 2017 | 2042 | 2048 | 2050 | 2076 | 0,988 | 0,985 | 0,984 | 0,972 | -10.064781 |

# Вывод

Использование автоматического распараллеливания оправдано не для всех задач. Потому что результаты полученные в ходе выполнения лабораторной работы показывают, что компилятор не смог провести автоматическое распараллеливание достаточной части программы, чтобы получить прирост скорости выполнения.