САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Дисциплина: Архитектура компьютера

Отчет

по лабораторной работе №4

«OpenMP»

Выполнил: Прудников Марк Олегович

20.Б09-мкн

Санкт-Петербург

2021

Цель работы: знакомство со стандартом OpenMP.

Инструментарий и требования к работе: C++. Стандарт OpenMP 2.0.

Теоретическая часть

OpenMP – C/C++ API, содержащее в себе директивы для компилятора, библиотечные функции и переменные окружения. Всё это используется для распараллеливания программ написанных на C/C++.

OpenMP использует fork-join модель параллельного исполнения. Программа, написанная с использованием OpenMP C/C++ API начинает своё исполнение как одиночный поток, который называется master thread. Master head исполняется последовательно в области, пока не встретится первая распараллеленная конструкция.

В своём решении я использовал директиву parallel for. Она является удобным сокращением для указания параллельной области, которая содержит в себе только одну конструкцию. То есть эта директива семантически имеет тот же смысл, что и директива parallel, которая определяет параллельную область, то есть область программы, которая должна исполняться несколькими потоками. Только в случае parallel for конкретно известно, что это единственная область, являющаяся циклом for.

Чтобы определить, какое количество потоков запрошено для выполнения выполнения области, используются следующие правила:

1. Если в директиве есть параметр num_threads, то он определяет количество запрошенных потоков.

2. Если была вызвана функция omp_set_num_threads, то переданный аргумент последней вызванной функции определяет количество потоков.
3. Если определена переменная окружения OMP_NUM_THREADS, то количество запрашиваемых потоков равняется значению этой переменной.
4. Если ни один из предыдущих пунктов не выполняется, то используется количество потоков по умолчанию.

Практическая часть

Количество потоков (первый аргумент) я устанавливал с помощью функции omp_set_num_threads.

В своём коде я распараллеливал циклы for с помощью прагмы #pragma parallel for.

Для выполнения замеров пункта “a” я использовал #pragma parallel for schedule(dynamic, img_sz / 4) и то же самое, только с параметром static, а количество потоков менялось (см. рисунок 1.1 и 1.2).

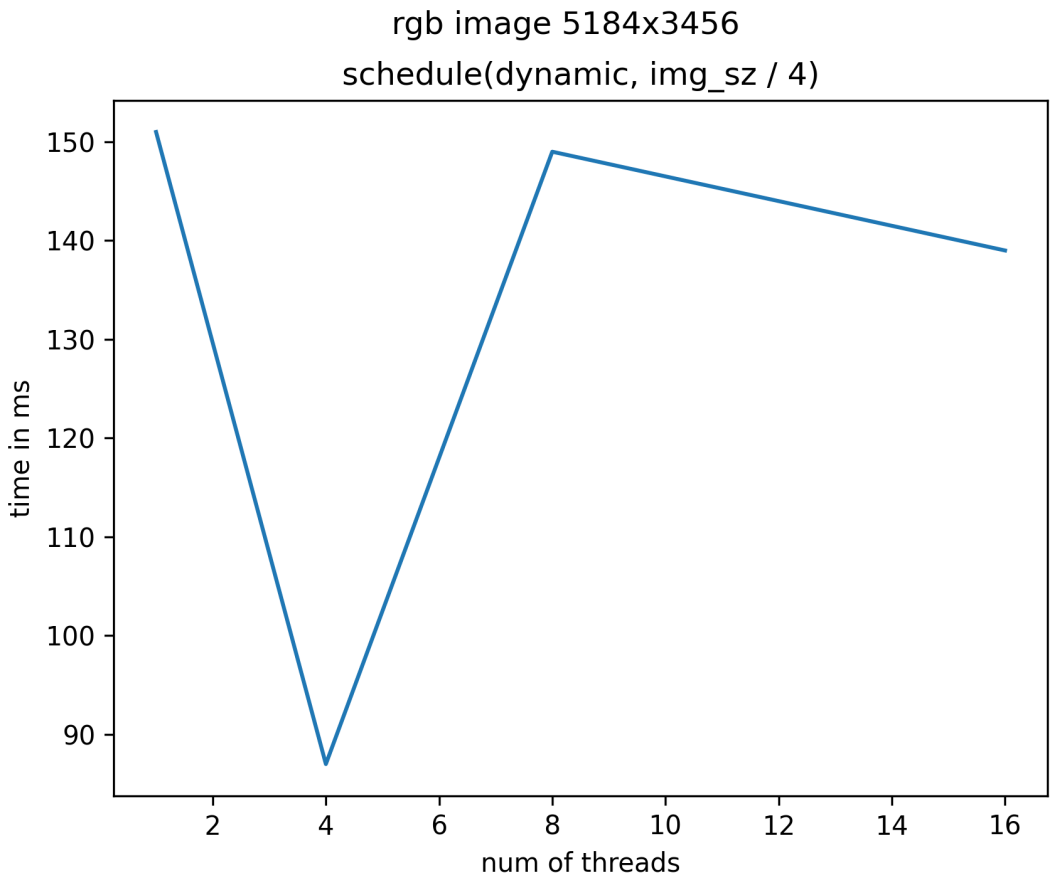

Рисунок №1.1 – Пункт а для dynamic

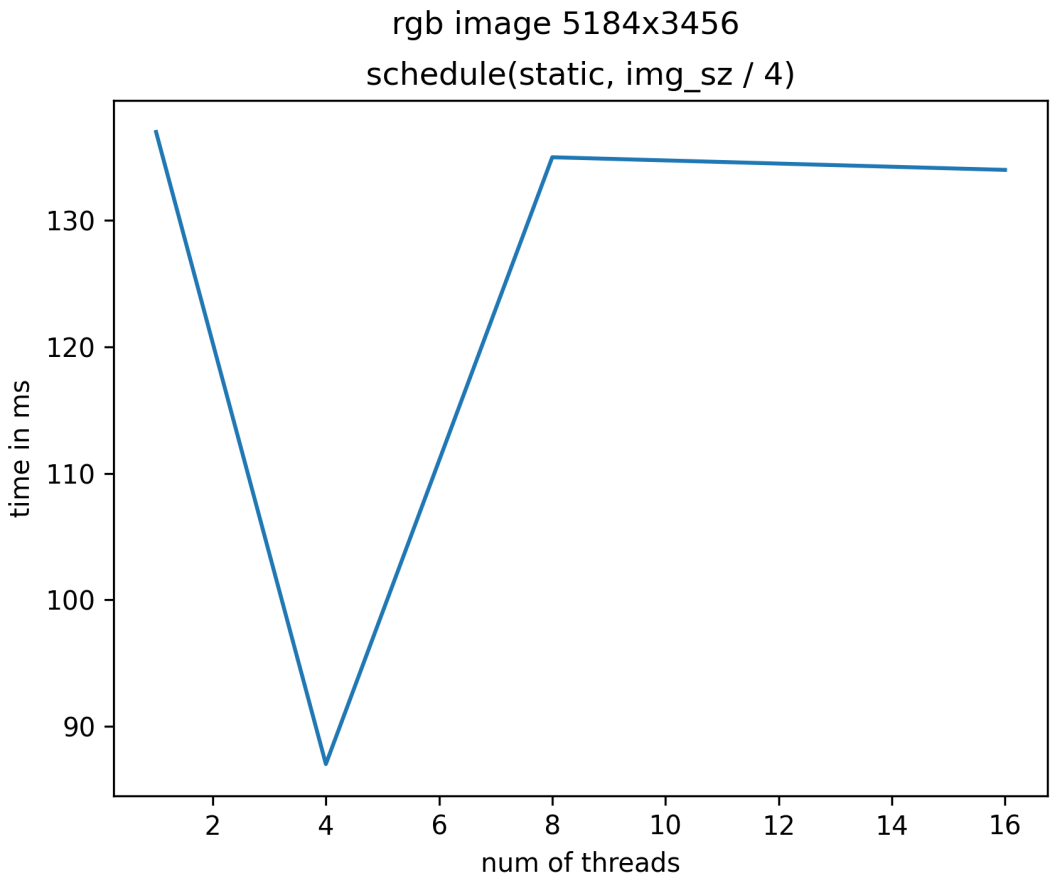

Рисунок №1.2 – Пункт a для static

Для выполнения замеров пункта “b” я использовал #pragma parallel for schedule и менял её параметры (см. рисунок 2), количество потоков равняется 4.

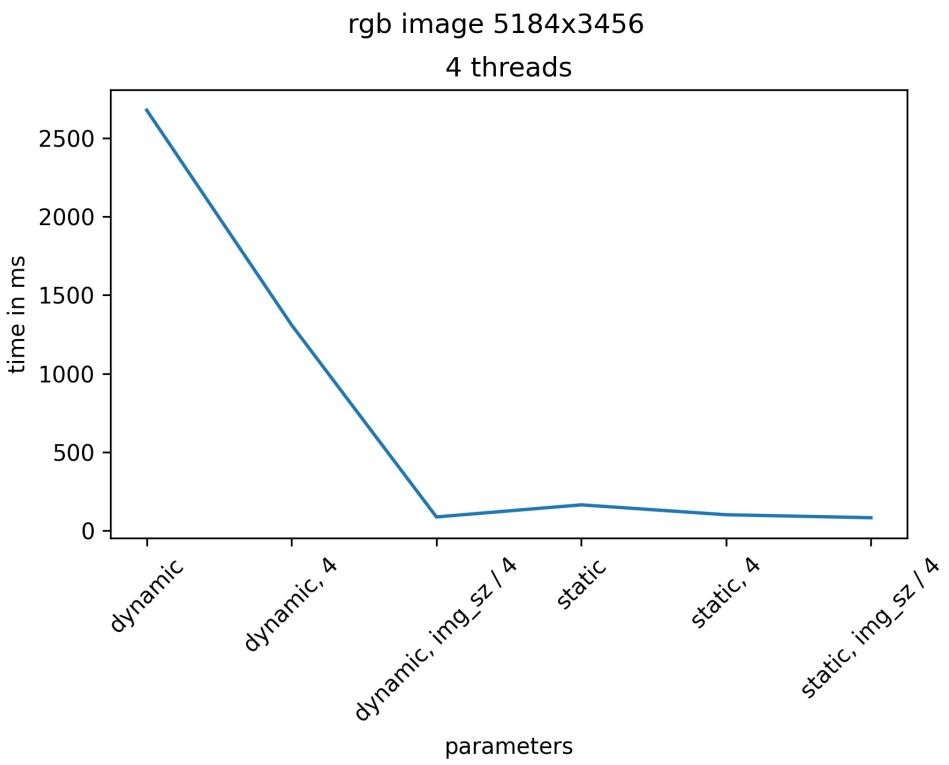

Рисунок №2 – Пункт b

Для выполнения замеров пункта “c” я закомментировал весь код, отвечающий за использование OpenMP. А потом вернул, и в качестве аргумента передал количество потоков равное 1 (см. рисунок 3).

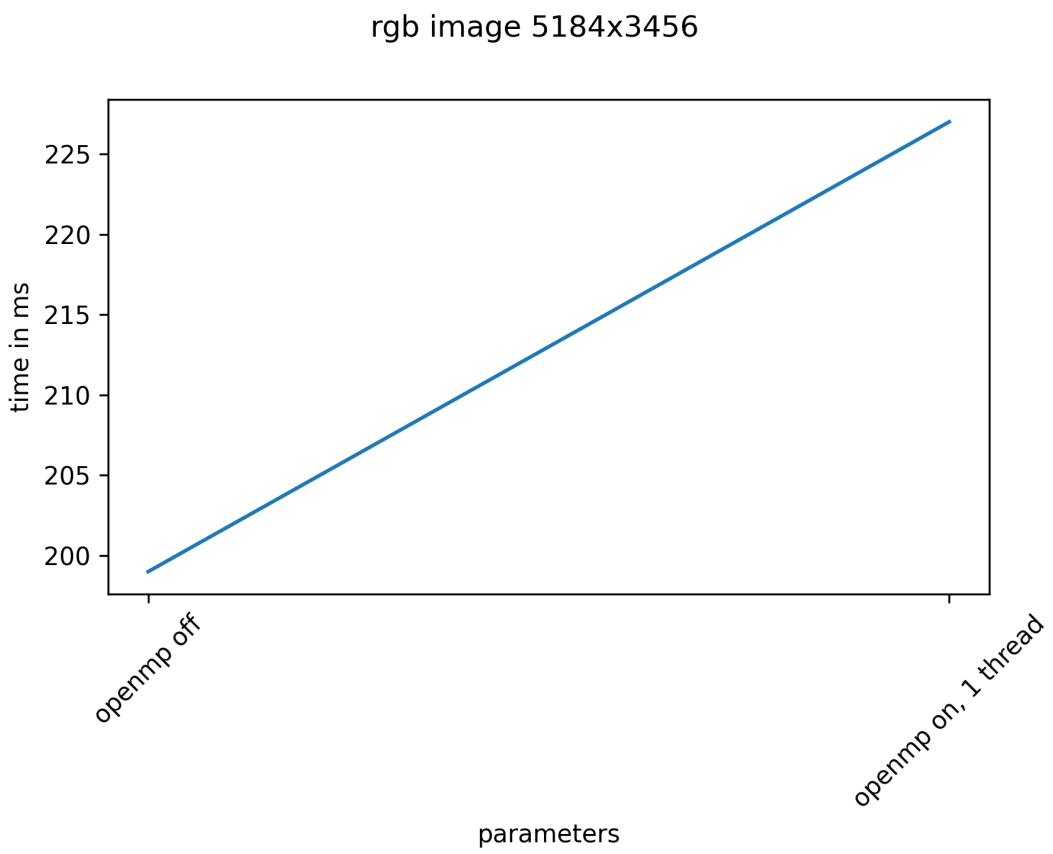

Рисунок №3 – Пункт c

Листинг

https://github.com/markprudnikov/aks-fall-21/tree/main/lab_04