Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Уфимский университет науки и технологий»

Кафедра ВМиК

Отчёт по лабораторной работе №1

на тему: «параллельное вычисление суммы ряда средствами автоматического распараллеливания и OpenMP»

по дисциплине: «Параллельные вычисления»

Выполнили:

Студент группы: ПРО-428Б Мохаммед А. А.

Аль-Шаибани Е. Т.

Вахитов Т. Ф.

Проверила: Шерыхалина Н.М.

Уфа – 2023

1. **Цель работы**

Научиться использовать основные директивы, функции и переменные окружения интерфейса OpenMP на примерах написания простейших параллельных программ для многоядерных вычислительных систем.

1. **Выполнение работы**
2. Выполнить программную реализацию на языке С / С++ последовательного алгоритма суммирования функционального ряда согласно варианту. Предусмотреть вывод на экран времени работы программы.
3. Произвести распараллеливание программы путём добавления директив OpenMP в текст последовательной программы.
4. Отладить написанные программы при небольшом числе членов ряда. Убедиться, что результаты последовательной и параллельной программ корректны и стабильны при многократных запусках. Убедиться, что количество порождаемых потоков равно числу ядер вычислительной системы.
5. Добавить опцию задания числа порождаемых потоков и запустить программу при различном числе потоков.
6. Подключить автоматическое распараллеливание к последовательной версии программы (ключ /QParallel). Убедиться, что количество порождаемых потоков равно числу ядер вычислительной системы.

A black text on a white background

Description automatically generated

A white sheet with black text

Description automatically generated

1. **Ход работы**

Задание 1:  
Дан ряд, согласно варианту 2

A math equations and symbols

Description automatically generated with medium confidence

**3. Сравнение параллельного и последовательного вычислений**

Последовательный и параллельный алгоритмы сравниваются для разных значений N. Результаты демонстрируют правильность распараллеливания и дают представление о достигнутом повышении производительности.A screenshot of a graph

Description automatically generated

**4. Вариация количества потоков**

Программа выполняется с различным количеством потоков, чтобы оценить масштабируемость параллельного алгоритма. Представлены результаты и проанализировано влияние различного количества потоков.

A screenshot of a computer screen

Description automatically generated

**6. Производительность для различный значений N**

Производительность как последовательных, так и параллельных алгоритмов оценивается для разных значений N. Результаты дают понимание того, как алгоритм масштабируется в зависимости от объёма задачи.A screenshot of a computer program

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| программа |  | Tp, c |  |
|  | N1 | N2 | N3 |
| Послед. | 26.739 | 53.97 | 113.769 |
| OpenMP | 4.771 | 9.476 | 20.735 |

**8. Стратегии балансировки загрузки**

Программа выполняется с различными стратегиями балансировки нагрузки (статическими, динамическими, управляемыми) и размерами блоков. Представлены результаты и обсуждается влияние на производительность.A screenshot of a computer

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Kind |  | TP , c |  |
|  | C1 | C2 | C3 |
| Статический | 5.288 | 5.053 | 5.023 |
| Динамический | 102.089 | 5.498 | 5.337 |
| Управляемый | 5.238 | 5.405 | 5.261 |

1. **Вывод**

В ходе выполнения данной лабораторной работы мы научились

использовать основные директивы, функции и переменные окружения интерфейса OpenMP на примерах написания простейших параллельных программ для многоядерных вычислительных систем.