Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики

Звіт

Лабораторна робота №3

Тема: «Розв’язування системи лінійних алгебр. рівнянь»
з дисципліни "Паралельні та розподілені обчислення"

Виконав студент групи ПМі-31
Процьків Назарій

Львів 2023 р.

Мета: написати програми обчислення розв’язку СЛАР послідовно та паралельно.

Хід роботи

Завдання виконав мовою програмування Python у середовищі PyCharm. Написав програму для розв’язування систем лінійних алгебраїчних рівнянь методом Крамера:

```
class Matrix:
    def __init__(self, height):...

    def __str__(self):...

    1 usage
    def generate(self):...

    1 usage
    def cramer_method_simple(self):...

    1 usage
    @staticmethod
    def worker(args, result, lock):...

    1 usage
    def cramer_method_parallel(self, num_threads):...
```

В main() згенерував матрицю 3 на 4 і обчислив двома методами, для того, щоб подивитись чи результат збігається.

```
if __name__ == '__main__':
    matrix = Matrix(3)
    matrix.generate()
    print(matrix)
    start_ = time.time()
    print(matrix.cramer_method_simple())
    end_ = time.time()
    print(matrix.cramer_method_parallel(2))
```

```
-4 0 8 7

-2 3 6 -5

-2 4 -9 -3


[-2.36702128 -2.62765957 -0.30851064]

[-2.36702128 -2.62765957 -0.30851064]
```

Також перевірив результат вручну.

Далі згенерував матрицю 200 на 201, заміряв час послідовного алгоритму, вивів його на екран, для кількості потоків від 2 до 10 провів по 2 тести обчислення і взяв з них середнє арифметичне. Вивів на екран всі результати:

```python
if __name__ == '__main__':
    matrix = Matrix(200)
    matrix.generate()

    start_ = time.time()
    matrix.cramer_method_simple()
    end_ = time.time()

    print(f"Dimensions: {(matrix.height, matrix.width)}")
    single_thread_time = end_ - start_
    print(f"Single thread time: {single_thread_time:.4f}s")
    threads = [2, 3, 4, 5, 6, 7, 8, 9, 10]

    for i in threads:
        current_time = []
        for _ in range(2):
            start_ = time.time()
            matrix.cramer_method_parallel(i)
            end_ = time.time()

            current_time.append(end_ - start_)

        current_time = sum(current_time) / 3
        acceleration = single_thread_time / current_time
        efficiency = acceleration / i

        print(f"Threads: {i} \t"
              f"Time: {current_time:.4f}s \t"
              f"Acceleration: {acceleration:.4f} \t"
              f"Efficiency: {efficiency:.4f}")
```

Результат:

```
Dimensions: (200, 201)
Single thread time: 1.4881s
Threads: 2  Time: 1.1136s   Acceleration: 1.3363    Efficiency: 0.6682
Threads: 3  Time: 0.8876s   Acceleration: 1.6765    Efficiency: 0.5588
Threads: 4  Time: 0.9011s   Acceleration: 1.6515    Efficiency: 0.4129
Threads: 5  Time: 1.3054s   Acceleration: 1.1399    Efficiency: 0.2280
Threads: 6  Time: 1.3982s   Acceleration: 1.0643    Efficiency: 0.1774
Threads: 7  Time: 0.9531s   Acceleration: 1.5613    Efficiency: 0.2230
Threads: 8  Time: 0.8371s   Acceleration: 1.7776    Efficiency: 0.2222
Threads: 9  Time: 0.8265s   Acceleration: 1.8004    Efficiency: 0.2000
Threads: 10     Time: 0.8908s   Acceleration: 1.6705    Efficiency: 0.1670
```

Висновок. під час виконання лабораторної роботи написав програму для обчислення СЛАР послідовним та паралельними алгоритмами, використовуючи метод Крамера, обчислив прискорення та ефективність для різної кількості потоків.