КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики

Звіт

до лабораторної роботи

з дисципліни «Хмарні обчислення»

на тему:

РОЗВ'ЯЗОК ЗАДАЧІ 2-SUM

З ВИКОРИСТАННЯМ СИСТЕМИ ПАРКС-PYTHON

Виконав студент 4-го курсу
кафедри теоретичної кібернетики
Некряч Владислав Вадимович

Київ – 2022

Постановка задачі

З використанням системи ПАРКС-Python реалізувати алгоритм розв’язку задачі 2-sum та проаналізувати прискорення, яке забезпечує використання кількох worker-ів.

Алгоритм

Нехай у нас є $n \geq 1$ воркерів. Розбиваємо вхідний масив на n частин, які порівну розподіляємо між воркерами, за допомогою методу `mymap`.

Фіксуємо елемент масиву a_i з виділеної частини, і для кожного такого елементу проходимо по масиву від елементу a_{i+1} до a_n в пошуках збігу.

Якщо сума збігається, додаємо два числа до масиву результатів даного воркера.

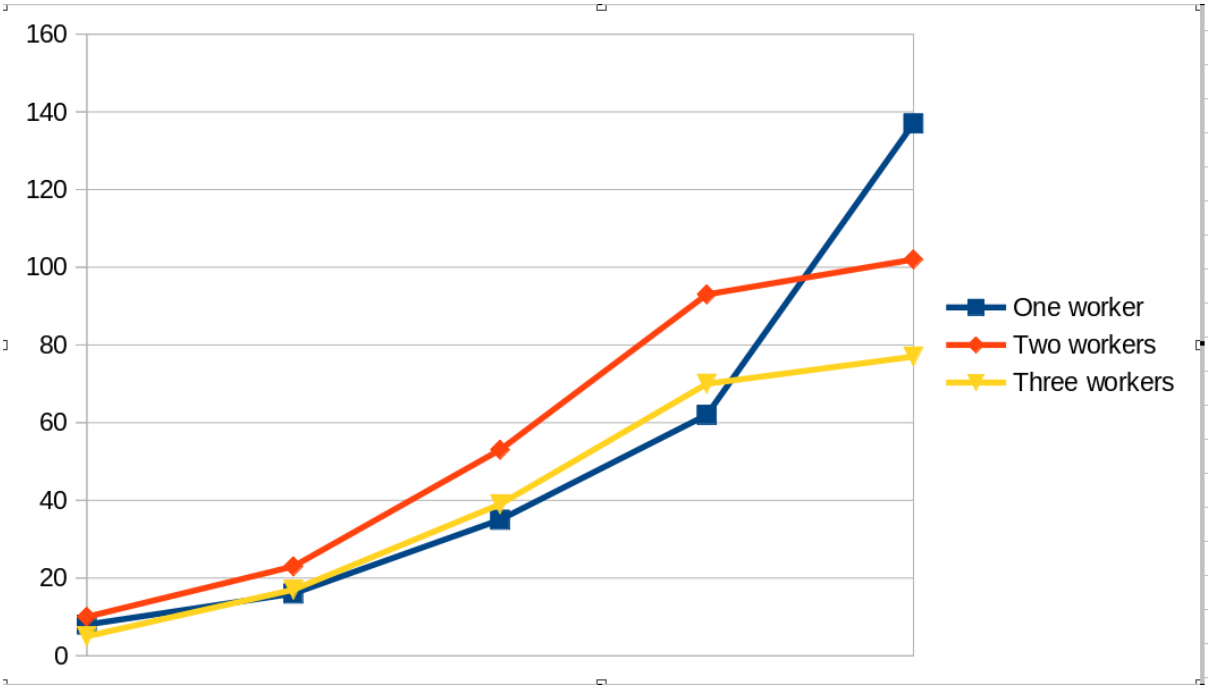
Наприкінці об'єднуємо масиви результатів воркерів в один за допомогою методу `myreduce` і повертаємо цей масив як фінальний результат.

```
    def solve(self):
        array, target = self.read_input()
        step = len(array) / len(self.workers)

        mapped = []
        for i in range(0, len(self.workers)):
            mapped.append(self.workers[i].mymap(i * step, min((i + 1) *
step, len(array) - 1), array, target))
        reduced = self.myreduce(mapped)
        self.write_output(reduced)

    @staticmethod
    @expose
    def mymap(a, b, array, target):
        res = []
        for fixed_element_index, fixed_element in enumerate(array[a:b +
1], a):
            if fixed_element_index + 1 < len(array):
                for moving_element in array[fixed_element_index + 1:]:
                    if int(fixed_element)+ int(moving_element) ==
int(target):
```

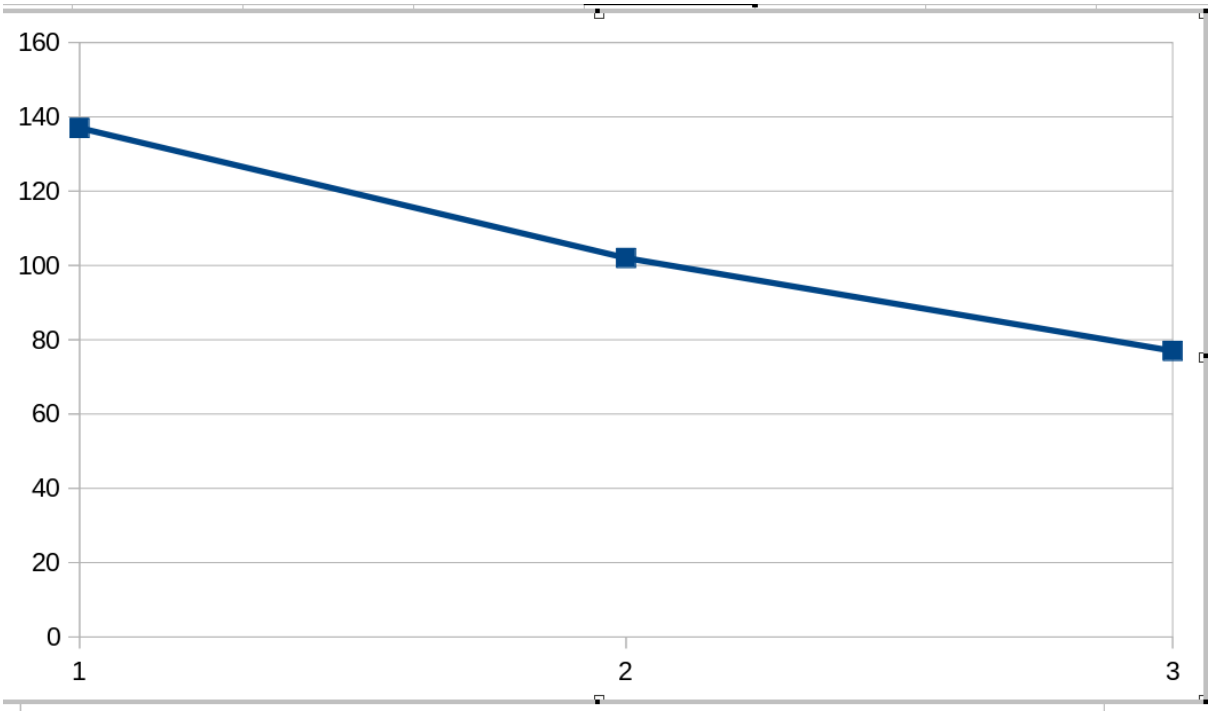
```
                        res.append([fixed_element, moving_element])
        return res


    @staticmethod
    @expose
    def myreduce(mapped):
        res = []
        for chunk in mapped:
            for s in chunk.value:
                res.append(s)
        return res
```

Результати виконання

1) Залежність часу виконання (с) від розміру вхідних даних *N (2.5K, 5K, 7.5K, 10K, 15K)*



2) Залежність часу виконання (с) від кількості воркерів для $N = 15K$



Висновок

Зі збільшенням розміру вхідних даних в усіх трьох випадках час виконання, як і очікувалося, зростає квадратично.

Використання кількох воркерів системи ПАРКС на даній задачі показало прискорення на великих розмірах масиву. Скоріш за все, це відбулося через високі накладні затрати на міжпроцесорне спілкування для більш малих задач.