

**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 MAX-BITWISE-OR  
З ВИКОРИСТАННЯМ СИСТЕМИ PARC-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 MAX-BITWISE-OR 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 елемента проходимо по масиву від елемента  $a_{i+1}$  до  $a_n$  в пошуках більшого результату.

Якщо поточний максимум менше нового результату, додаємо змінюємо максимум даного воркера.

Наприкінці об'єднуємо результати воркерів за допомогою методу `myreduce` і повертаємо максимум максимумів як фінальний результат.

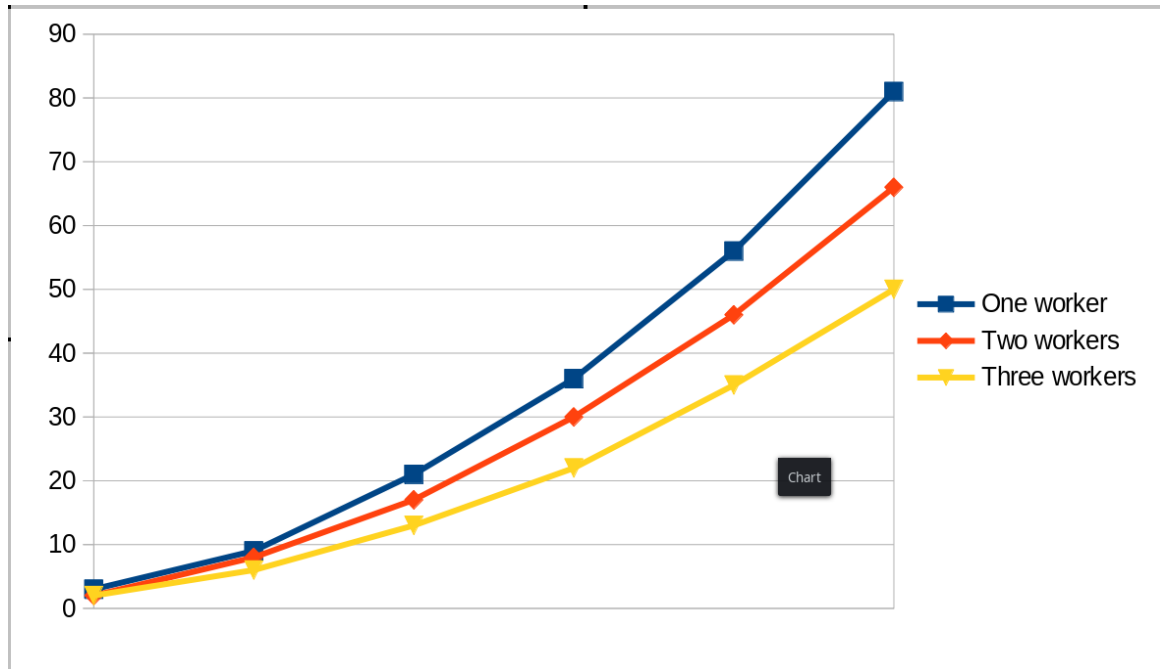
```
def solve(self):  
    array = self.read_input()  
    step = len(array) / len(self.workers)  
  
    mapped = []  
    for i in range(0, len(self.workers)):  
        mapped.append(self.workers[i].mymap(i * step, min((i + 1) *  
step, len(array) - 1), array))  
    reduced = self.myreduce(mapped)  
    self.write_output(reduced)  
  
    @staticmethod  
    @expose  
    def mymap(a, b, array):  
        max_result = float('-inf')  
        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) >
max_result:
        max_result = int(fixed_element) |
int(moving_element)
    return max_result

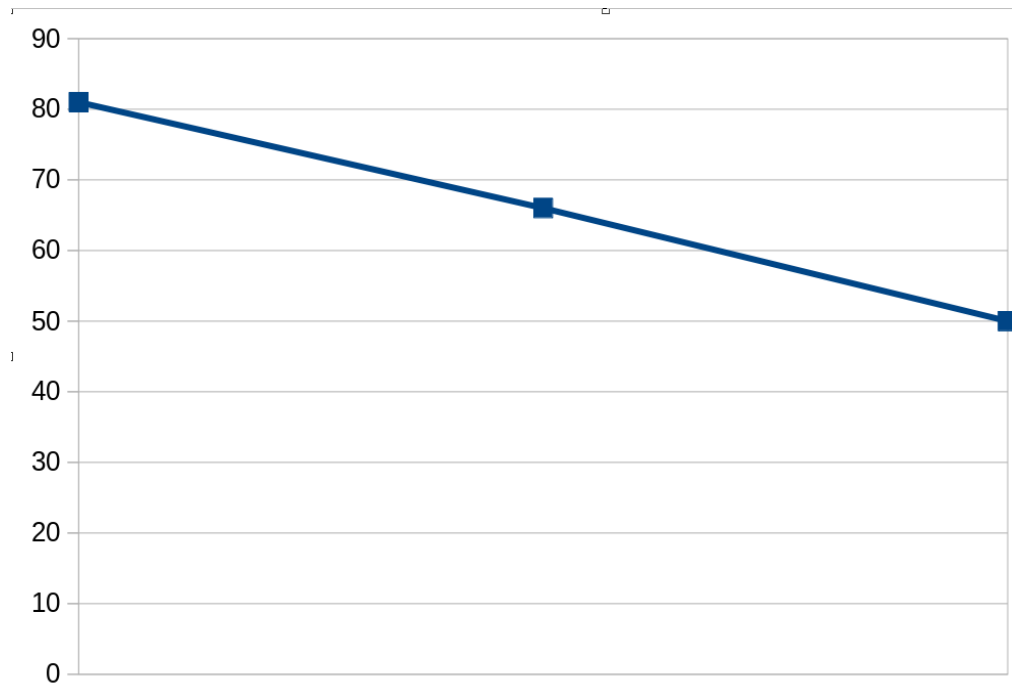
    @staticmethod
    @expose
    def myreduce(mapped):
        maximum = float('-inf')
        for chunk in mapped:
            if chunk.value > maximum:
                maximum = chunk.value
        return maximum
```

## Результати виконання

1) Залежність часу виконання ( $c$ ) від розміру вхідних даних  $N$  ( $2.5K$ ,  $5K$ ,  $7.5K$ ,  $10K$ ,  $12.5K$ ,  $15K$ )



2) Залежність часу виконання ( $c$ ) від кількості воркерів (1-3) для  $N = 15K$



## **Висновок**

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

Як і очікувалось, використання кількох воркерів системи ПАРКС на даній задачі показало прискорення із збільшенням кількості воркерів.