**Звіт до Лабораторної роботи 2**

**Потєхіной Валерії**

**Мета роботи:**  
Метою даної лабораторної роботи було дослідження продуктивності стандартного алгоритму std::replace\_if при послідовному та паралельному виконанні на різних обсягах даних. Робота передбачала порівняння ефективності алгоритму з різними політиками виконання (std::execution::seq, std::execution::par, std::execution::par\_unseq), аналіз впливу кількості потоків на швидкодію, а також розробку та тестування власної паралельної реалізації алгоритму для визначення оптимального числа потоків K у порівнянні з апаратними можливостями процесора.  
   
**Хід роботи**:  
Експериментальна робота проводилася на наборі даних різної довжини. Для кожного набору даних проводилося:

1. Послідовне виконання алгоритму replace\_if без політики.
2. Виконання з політиками std::execution::seq, std::execution::par та std::execution::par\_unseq.
3. Виконання власної паралельної реалізації алгоритму з різною кількістю потоків K.

Для власного алгоритму обчислювався час виконання для кожного K, а також визначався оптимальний K і його співвідношення з кількістю апаратних потоків процесора.

**Результати експериментів:**  
   
**Система:**8 апаратних потоків.

**Таблиця 1. Час виконання (мс) для 1,000,000 елементів**

|  |  |
| --- | --- |
| **Алгоритм / Політика** | **Час виконання (ms)** |
| Послідовний (без політики) | 5.86 |
| std::execution::seq | 6.61 |
| std::execution::par | 3.06 |
| |  | | --- | | std::execution::par\_unseq |  |  | | --- | |  | | 1.60 |
| |  | | --- | | Власний (Best K = 4) |  |  | | --- | |  | | 1.99 |

Таблиця 2. Час виконання (мс) для 5,000,000 елементів

|  |  |  |  |
| --- | --- | --- | --- |
| **Алгоритм / Політика** | | **Час виконання (ms)** | | --- |  |  | | --- | |  | |
| |  | | --- | | Послідовний (без політики) |  |  | | --- | |  | | 28.67 |
| |  | | --- | | std::execution::seq |  |  | | --- | |  | | 30.75 |
| |  | | --- | | std::execution::par |  |  | | --- | |  | | 6.66 |
| |  | | --- | | std::execution::par\_unseq |  |  | | --- | |  | | 5.54 |
| |  | | --- | | Власний (Best K = 8) |  |  | | --- | |  | | 5.93 |

**Аналіз залежності часу власного алгоритму від**K**показав,** що найкраща швидкодія досягається при K = 4 для 1,000,000 елементів і при K = 8 для 5,000,000 елементів. Співвідношення K/hardware\_threads становить 0.5 та 1 відповідно.

**Аналіз результатів та висновки:**

1. Ефективність паралельного виконання алгоритму replace\_if прямо залежить від обсягу даних. Для невеликих наборів даних виграш від паралелізму мінімальний, а для великих - значний.
2. Найкраща кількість потоків K для власної реалізації зазвичай дорівнює кількості апаратних потоків процесора. При подальшому збільшенні K час виконання може зростати через накладні витрати на керування потоками.
3. Стандартні політики std::execution::par і par\_unseq демонструють високу ефективність та майже не поступаються власній реалізації, що робить їх зручними для практичного застосування.
4. Дослідження підтвердило, що розумне використання паралельності дозволяє значно прискорити виконання ресурсоємних операцій, однак для малих обсягів накладні витрати на створення потоків можуть переважити виграш у швидкодії.