

Висновки дослідження

У роботі досліджено швидкодію алгоритму `std::count_if` при різних режимах:

1. Послідовний алгоритм
2. Паралельні політики стандартної бібліотеки (`std::execution::par`, `par_unseq`)
3. Власна реалізація паралельного алгоритму з параметром K , який визначає кількість підділянок для обробки.

Дослідження проводилися для трьох розмірів даних:

100 тис., 1 млн, 5 млн елементів, з двома предикатами - легким та важким.

Основні результати:

- Паралельні політики стандартної бібліотеки забезпечують прискорення у 4–6 разів для легких операцій і до 7 разів для важких.
- Власний алгоритм демонструє залежність часу виконання від параметра K .
При малих K недовикористовуються ресурси процесора.
При надто великих K час зростає через накладні витрати синхронізації та створення потоків.
- Оптимальні значення K :
 - Для $N=100k \rightarrow K = 4$
 - Для $N=1M \rightarrow K = 8$
 - Для $N=5M \rightarrow K = 8$ (легкий), $K = 16$ (важкий)
- Значення K , які дають найкращий час, знаходяться приблизно в межах $K \approx 1/3 \dots 1.5 \times (\#CPU\ Threads)$.

Висновок:

Паралельні політики бібліотеки C++ є значно ефективнішими та простими у використанні, однак власний алгоритм також демонструє хорошу масштабованість при коректному виборі параметра K .