

В цій лабораторній роботі потрібно було дослідити ефективність роботи операції `replace_if` над послідовностями цілих чисел, порівнюючи її виконання використовуючи алгоритми без політики, з політиками (послідовна, паралельна, паралельно-невпорядкована) і власний паралельний алгоритм. Порівнюючи швидкість роботи алгоритму з різними політиками, то найкраще проявляє себе паралельна, так як вона використовує кілька потоків одночасно. Паралельно-невпорядкована політика інколи демонструє ще менший час за рахунок додаткової векторизації, але не завжди суттєво краще для невеликих обсягів. Власний паралельний алгоритм показав, що час виконання залежить від числа розбитих частин K . При малому K частини досить великі, тому частина ядер процесора може залишатися недовантаженою, і час виконання не скорочується суттєво. Зі збільшенням K навантаження розподіляється більш рівномірно між потоками, що дозволяє повністю задіяти всі ядра і скоротити час обробки. Надто велике K призводить до збільшення часу виконання через накладні витрати на створення великої кількості потоків і синхронізацію їх роботи. Найкраща швидкість досягається, коли K приблизно дорівнює кількості фізичних або логічних ядер процесора, або трохи перевищує її. Це підтверджує ефективність алгоритму при збалансованому розподілі навантаження. Режим `release` з максимальною оптимізацією дав значно менший час виконання порівняно з режимом без оптимізації.

Отже, експерименти показали, що ефективність паралельних алгоритмів значною мірою залежить від числа потоків, обсягу даних та рівня оптимізації компілятора. Власний паралельний алгоритм дозволяє досягти максимального прискорення, якщо число частин K +- рівне кількості фізичних або логічних ядер процесора.