

Іванова Ангеліна К-27

У ході лабораторної роботи ми дослідили поведінку стандартного алгоритму `adjacent_difference` для обчислення максимальної різниці між сусідніми елементами у послідовності. Для цього ми згенерували кілька наборів випадкових даних різної довжини, використовуючи засоби `std::random`, після чого послідовно вимірювали час роботи алгоритмів у різних режимах.

Спочатку ми протестували звичайний бібліотечний алгоритм без політик виконання, щоб отримати базове значення продуктивності. Далі виконали ті самі вимірювання, але вже з використанням паралельних політик (`std::execution::seq, par, par_unseq`). Це дозволило порівняти, як саме політики впливають на швидкодію, і чи дають вони прискорення для конкретного типу операції. З отриманих результатів видно, що паралельні політики можуть давати виграш, але не завжди — все залежить від обсягу даних і характеру операції над ними.

Після цього ми реалізували власний паралельний алгоритм, який розбиває послідовність на K приблизно однакових частин, обробляє кожну частину у власному потоці, а потім об'єднує результати. Ми запускали цей алгоритм для різних значень K, вимірювали час виконання та будували таблицю залежності продуктивності від кількості потоків. Це дозволило визначити, при якому K досягається найшвидша робота, і порівняти це значення з реальною кількістю апаратних потоків у процесорі. У результаті стало помітно, що найкраща швидкість досягається тоді, коли кількість частин приблизно відповідає кількості доступних апаратних ядер або трохи перевищує їх. За надто великих значень K час починає зростати через накладні витрати на створення потоків і зведення результатів.

Окремо ми протестували, як на швидкодію впливає рівень оптимізації компілятора: без оптимізації час виконання був найбільшим, а в режимі максимальних оптимізацій (Release) результати були суттєво швидшими. Це підтвердило, що оптимізації компілятора мають значний вплив на продуктивність алгоритмів роботи з великими масивами даних.

У підсумку ми отримали повну картину того, як стандартні й власноруч реалізовані алгоритми працюють із різними наборами даних і параметрами паралельності. Робота дала можливість порівняти підходи, побачити їхні сильні та слабкі сторони та зробити висновок, що оптимальний вибір

залежить і від розміру даних, і від апаратних характеристик системи, і від того, наскільки складною є операція, яку виконує алгоритм.