

Звіт

Варіант 6 — скалярний добуток (`inner_product`) (з політиками використовувати `transform_reduce`)

Дубас Дмитро К-25

Мета:

Дослідити швидкодію стандартного алгоритму `std::inner_product` за відсутності політики та `std::transform_reduce` з різними політиками виконання (`seq`, `par`, `par_unseq`), а також власну реалізацію паралельного `inner_product` для різної кількості потоків `K`.

Хід роботи:

1. Згенеровано випадкові послідовності довжин 1000000, 10000000, 100000000.
2. Виконано обчислення скалярного добутку двох векторів (операція `inner_product`).
3. Для кожного випадку виміряно час виконання:

- без політики (`inner_product`);

- з політиками `seq`, `par`, `par_unseq` (`transform_reduce`);

- власний паралельний алгоритм при різному `K` (від 1 до 32).

Результати:

$N = 1\,000\,000$:

- inner_product = 12.81 мс, seq = 3.87 мс, par = 6.36 мс, par_unseq = 2.46 мс

Власний паралельний алгоритм: найкраще $K = 3$, час = 7.74 мс, апаратні потоки = 12

$N = 10\,000\,000$:

- inner_product = 270.69 мс, seq = 34.36 мс, par = 18.67 мс, par_unseq = 19.56 мс

Найкраще $K = 7$, час = 39.20 мс, апаратні потоки = 12

$N = 100\,000\,000$:

- inner_product = 1077.96 мс, seq = 805.79 мс, par = 133.64 мс, par_unseq = 221.33 мс

Найкраще $K = 3$, час = 296.55 мс, апаратні потоки = 12

Висновки:

1. При збільшенні кількості потоків до значення, близького до половини кількості апаратних

потоків процесора (3–7 з 12), спостерігається мінімальний час виконання.

2. Подальше збільшення K (понад 12) призводить до зростання накладних витрат на створення та синхронізацію потоків (лінійне зростання $\sim 10\text{--}58$ мс на потік).

3. Алгоритм `std::transform_reduce(execution::par)` показує кращу продуктивність при великому N , тоді як `par_unseq` — при малому (завдяки векторизації).

4. Для коротких послідовностей переваги паралелізації обмежені `overhead`, для довгих —
значні (прискорення до 6–8 разів).

5. Режим Release з оптимізацією (-O3) істотно зменшує час виконання (у 2–10 разів порівняно з -O0) за рахунок векторизації та інлайнінгу.