

Звіт

Методологія дослідження

Дослідження проводилося на алгоритмах трансформації з використанням різних політик виконання: стандартної послідовної, паралельної та паралельної з векторизацією. Тестування включало дві типи операцій - легкі ("швидкі") та складні ("повільні") обчислення на наборах даних розміром від 1 тисячі до 1 мільйона елементів. Додатково досліджувалась оптимізація кількості потоків виконання.

Ключові результати та аналіз

Для малих обсягів даних (до 10 тисяч елементів) паралелізація виявилася неефективною через значні накладні витрати на створення потоків. Однак для великих масивів даних від 500 тисяч елементів спостерігається значне прискорення - у 4-6.5 разів для складних операцій. Найкращу продуктивність демонструє політика ``parallel_unsequenced``, яка поєднує паралелізм із векторизацією.

Цікавим відкриттям стало те, що оптимальна кількість потоків перевищує кількість апаратних ядер процесора у 2.8-3.8 рази, що особливо виражено для наборів даних розміром 100 тисяч елементів (оптимальне $K=45$). Для мільйона елементів найкращий результат досягається при $K=34$ із прискоренням майже у 7 разів.

Основні висновки:

- Паралельні алгоритми ефективні лише для великих масивів даних (від 100K елементів)
- Складні операції отримують більшу вигоду від паралелізації
- Оптимальна кількість потоків залежить від розміру даних
- Політика ``parallel_unsequenced`` є найбільш універсальним вибором

Переваги паралелізації підтверджені експериментально:

- Прискорення до 6.99x для великих даних
- Ефективне використання багатоядерних архітектур
- Можливість оптимізації під конкретні розміри даних