

# Звіт про лабораторну роботу №4

з дисципліни “Кластерні розрахунки” (High-Performance Computing)

студента 1 курсу групи ПЗС-1

Грищенко Юрія

## Тема: Паралельний алгоритм сортування даних

Робота виконана з використанням бібліотеки Intel MPI на ОС Linux.

(див. відео з Google Drive)

### Результати

Array size	Serial Bubble	Parallel Bubble				std::sort
		2 processors		4 processors		
		Time	Speedup	Time	Speedup	
10	0.000003	0.000666	-0.000663	0.004505	-0.003839	0.000006
100	0.000045	0.002161	-0.002116	0.006196	-0.004035	0.000017
10000	0.392749	0.550864	-0.158115	0.482922	0.067942	0.002173
20000	1.585668	2.193885	-0.608217	1.962906	0.230979	0.004261
30000	3.575783	5.013062	-1.437279	4.28358	0.729482	0.007022
40000	6.471324	8.90495	-2.433626	7.641064	1.263886	0.009096
50000	10.031491	13.914632	-3.883141	12.263332	1.6513	0.01219

Array size	2 processors		4 processors	
	Model	Experiment	Model	Experiment
10	3.5679313965698E-07	0.000666	2.0812933146657E-07	0.004505
100	2.1407588379419E-05	0.002161	6.54120756037802E-06	0.006196
10000	0.198376985649282	0.550864	0.049713177458873	0.482922
20000	0.793190793139657	2.193885	0.198535560378019	1.962906
30000	1.78444142247112	5.013062	0.446467148757438	4.28358
40000	3.17212887364368	8.90495	0.79350794259713	7.641064
50000	4.95625314665733	13.914632	1.2396579418971	12.263332

## **Висновки**

- Для вирішення задачі використали бульбашкове сортування
- Ознайомилися з принципом парного-непарного сортування
- Імплементували послідовний та паралельний алгоритми для вирішення задачі
- Дослідили час виконання алгоритмів над вхідними даними різного розміру
- Теоретично оцінили час виконання паралельного алгоритму, порівняли з дійсними результатами
- Отримали неочікуваний результат, що нагадує лабораторну роботу №1: послідовний алгоритм швидший за паралельний. Реальний час виконання значно перевищує теоретичний.
  - Це може свідчити про високу ефективність кешу процесора при роботі (досить простого) послідовного алгоритму над простою лінійною структурою даних, за рахунок чого отримуємо пришвидшення в порівнянні зі складнішим паралельним алгоритмом.
  - Бачимо що `std::sort` має кращу часову складність, отже працює значно швидше за бульбашкове сортування, навіть якщо воно розподілено на 4 процесорах.