

# Звіт про лабораторну роботу №2

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

студента 1 курсу групи ПЗС-1

Грищенко Юрія

## Тема: Паралельний алгоритм множення матриці на матрицю

Робота виконана з використанням бібліотеки Intel MPI на ОС Linux.

(див. відео з Google Drive)

### Результати

Matrix size	Serial	Parallel	
		4 processors	
		Time	Speedup
10	0,00002	0,00036	-0,00034
100	0,01102	0,00623	0,00479
500	2,26221	0,50225	1,75996
1000	12,00000	4,62900	7,37100
1500	49,60608	20,65509	28,95098
2000	104,29314	30,15918	74,13396
3000	365,00000	108,00000	257,00000

Matrix size	4 processors	
	Model	Experiment
10	5,7028514257129E-07	0,00004
100	5,97298649324662E-05	0,00014
500	0,00149924962481241	0,00679
1000	0,006	0,02460
2000	0,0240060030015007	0,05437
3000	0,0540180090045023	0,09939

## **Висновки**

- Визначили задачу множення матриці на матрицю
- Імплементували послідовний та паралельний алгоритми для вирішення задачі
- Ознайомилися з картезіанською (табличною) топологією, функціями MPI\_Cart\_\*, використанням кількох комунікаторів
- Дослідили час виконання алгоритмів над вхідними даними різного розміру
- Теоретично оцінили час виконання паралельного алгоритму, порівняли з дійсними результатами
- Отримали очікуваний результат: паралельний алгоритм швидший за послідовний.
  - В порівнянні з лабораторною роботою №1 послідовний алгоритм виконується над складнішою структурою даних, яка не чітко відображається лінійно в оперативній пам'яті, отже кеш процесора відіграє набагато меншу роль.
  - Також можливо, що паралельний алгоритм дає очікуваніший результат (в порівнянні з лабораторною №1) через заміну OpenMPI на Intel MPI.
  - При цьому все ж бачимо, що результат на практиці приблизно вдвічі повільніший за теоретичний. Причиною може бути складність топології при паралелізації даного алгоритму і спричинені цим затримки (синхронізації процесів)