

Звіт про лабораторну роботу №1

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

студента 1 курсу групи ПЗС-1

Грищенко Юрія

Тема: Паралельний алгоритм множення матриці на вектор

Робота виконана з використанням бібліотеки OpenMPI на ОС Linux.

(див. відео з Google Drive)

Результати

Matrix size	Serial	Parallel			
		2 processors		4 processors	
		Time	Speedup	Time	Speedup
10	0,00000	0,00004	-0,00003	0,00017	-0,00013
100	0,00007	0,00014	-0,00007	0,00019	-0,00005
1000	0,00553	0,00679	-0,00126	0,00896	-0,00217
2000	0,02104	0,02460	-0,00356	0,02810	-0,00350
3000	0,04568	0,05437	-0,00869	0,06127	-0,00691
4000	0,08038	0,09939	-0,01901	0,10787	-0,00848
5000	0,12894	0,15051	-0,02156	0,16814	-0,01763
6000	0,18432	0,21705	-0,03272	0,24344	-0,02639
7000	0,24329	0,30108	-0,05779	0,32883	-0,02775
8000	0,32622	0,38695	-0,06073	0,43442	-0,04747
9000	0,41058	0,51650	-0,10592	0,55068	-0,03418
10000	0,53517	0,66625	-0,13109	0,74795	-0,08170

Matrix size	2 processors		4 processors	
	Model	Experiment	Model	Experiment
10	2,5421608580429E-07	0,00004	1,5252965148258E-07	0,00017
100	2,6625790039502E-05	0,00014	1,3312895019751E-05	0,00019
1000	0,00267462081854093	0,00679	0,00133731040927046	0,00896

2000	0,0107011592329616	0,02460	0,00535057961648082	0,02810
3000	0,0240796152432622	0,05437	0,0120398076216311	0,06127
4000	0,0428099888494425	0,09939	0,0214049944247212	0,10787
5000	0,0668922800515026	0,15051	0,0334461400257513	0,16814
6000	0,0963264888494425	0,21705	0,0481632444247212	0,24344
7000	0,131112615243262	0,30108	0,0655563076216311	0,32883
8000	0,171250659232962	0,38695	0,0856253296164808	0,43442
9000	0,216740620818541	0,51650	0,10837031040927	0,55068
10000	0,2675825	0,66625	0,13379125	0,74795

Висновки

- Визначили задачу множення матриці на вектор
- Імплементували послідовний та паралельний (з використанням OpenMPI) алгоритми для вирішення задачі
- Ознайомилися з базовими функціями MPI_Init, MPI_Comm_*, MPI_Scatterv, MPI_Allgatherv тощо
- Дослідили час виконання алгоритмів над вхідними даними різного розміру
- Теоретично оцінили час виконання паралельного алгоритму, порівняли з дійсними результатами
- Отримали неочікуваний результат: послідовний алгоритм швидший за паралельний. Можливі фактори:
 - Це може свідчити про неефективність роботи OpenMPI або його використання в нашій конкретній програмі (зокрема визначили що крок DataDistribution займає >80% часу виконання програми)
 - Така неефективність є дивною, оскільки всі процеси виконуються на одній машині зі спільною пам'яттю, отже не повинно бути затримки через надсилання даних.
 - Це може свідчити про занадто довге очікування синхронізації процесів
 - В наступній лабораторній буде використано Intel MPI
 - Також 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