Pertea Alexandra-Ioana
Grupa 235/2

Laborator 3 PPD

Deadline: săptămâna 7

1. Cerința problemei

Să se scrie un program bazat pe MPI care face suma a 2 numere mari. Un număr mare = număr cu mai mult de 10 cifre. Reprezentarea unui număr se face sub forma unui tablou de cifre (numere întregi fără semn), în care cifra cea mai nesemnificativă este pe prima poziție.

Cele 2 numere mari se citesc din fisierele “Numar1.txt” și “Numar2.txt”, iar rezultatul se afișează în fișierul “Numar3.txt”.

2. Proiectarea aplicației:

Varianta 0:

Se citesc cifrele din fișierele corespunzătoare și se adaugă în vectori, mereu pe prima poziție. Se calculează secvențial suma, iar apoi se afișează în fișierul corespunzător vectorul rezultat, parcurs în ordine inversă.

Varianta 1:

Procesul 0 citește câte N / p cifre și le trimite procesului curent. Celelalte procese calculează sumele și carry-ul corespunzător. Fiecare proces (cu excepția ultimului) trimite următorului carry-ul calculat, iar procesul 1 nu primește carry (îl consideră 0). Procesul 0 scrie rezultatul în fișier.

Modul de proiectare: procesele primesc carry înainte de a primi cifrele pe care trebuie să le adune.

Varianta 2:

Procesul 0 citește cele 2 numere și le stochează în 2 șiruri. Cifrele celor două numere se distribuie proceselor prin MPI_Scatter (câte N / p cifre), ele fac suma, trimit carry-ul (cu excepția ultimului), iar rezultatul final se obține în procesul 0 prin MPI_Gather. Procesul 0 scrie rezultatul în fișier.

3. Rezultatele testelor

Numerele	Varianta	Numărul de procese	Timp execuție (ms)
Număr1 = Număr2 = “123456789123456789”	Varianta 0	-	0.68584
	Varianta 1	4 procese	3.69218
		8 procese	6.67266
	Varianta 2	4 procese	1.94947
		8 procese	3.16408

N_1 = 1000 N_2 = 1000 (random digits)	Varianta 0	-	11.08198
	Varianta 1	4 procese	12.47022
		8 procese	16.87284
	Varianta 2	4 procese	11.63597
		8 procese	11.25906
N_1 = 100 N_2 = 100 000 (random digits)	Varianta 0	-	1029.118
	Varianta 1	4 procese	11870.38
		8 procese	12002.58
	Varianta 2	4 procese	11545.47
		8 procese	11609.60

4. Analiza rezultatelor

Varianta 2 e mai eficientă din punct de vedere al timpului de execuție.

Device-ul de pe care s-a testat are 8 procesoare – cu toate acestea, rularea cu 8 procese a fost mereu mai lentă decât cea cu 4 procesoare.