Haja Florin-Gabriel

Programare Paralelă și Distribuită – Laborator 3

Scrieti un program bazat pe MPI care face suma a 2 numere mari. ‘numar mare’ = numar cu mai mult de 10 cifre

Reprezentare = tablou de cifre (numere intregi fara semn - byte) in care cifra cea mai nesemnificativa este pe prima pozitie. Cele 2 numere mari se citesc din fisierele “Numar1.txt” si “Numar2.txt”. Fiecare din aceste fisiere contine la inceput un numar (N) care reprezinta numarul de cifre si apoi cifrele numarului respectiv. Implementare ->C++11.

Varianta 0 – implementare secventiala C++11. p - procese MPI

Varianta 1 – considera rezolvarea problemei prin executia urmatoarelor etape: 1) procesul 0 citeste cele 2 numere si le stocheaza in 2 tablouri: a. daca un numar are mai putine cifre se completeaza cu cifre nesemnificative

2) cifrele celor 2 numere se distribuire proceselor folosind MPI\_Scatter (daca nu este valabila conditia p|N, unde N=max{N\_1,N2}, N\_1 nr de cifre ale primului numar, N\_2 nr de cifre ale celui de-al doilea, atunci se mareste N corespunzator si se completeaza cu 0-uri)

3) procesele fac suma cifrelor primite si calculeaza “report” (carry) corespunzator 4) fiecare process (cu exceptia ultimului) trimit “reportul” la procesul urmator care il foloseste pentru actualizarea rezultatului

5) rezultatul final se obtine in procesul 0 (MPI\_Gather) 6) procesul 0 scrie rezultatul in fisierul “Numar3.txt”

Varianta2– considera rezolvarea problemei prin executia urmatoarelor etape: 1) id\_proces\_curent=1 2) procesul 0 repeta urmatoarea actiune pana cand se citesc toate cifrele numerelor a. citeste cate n/p cifre din cele 2 fisiere b. le rimite procesului “id\_proces\_curent” c. incrementeaza “id\_proces\_curent”

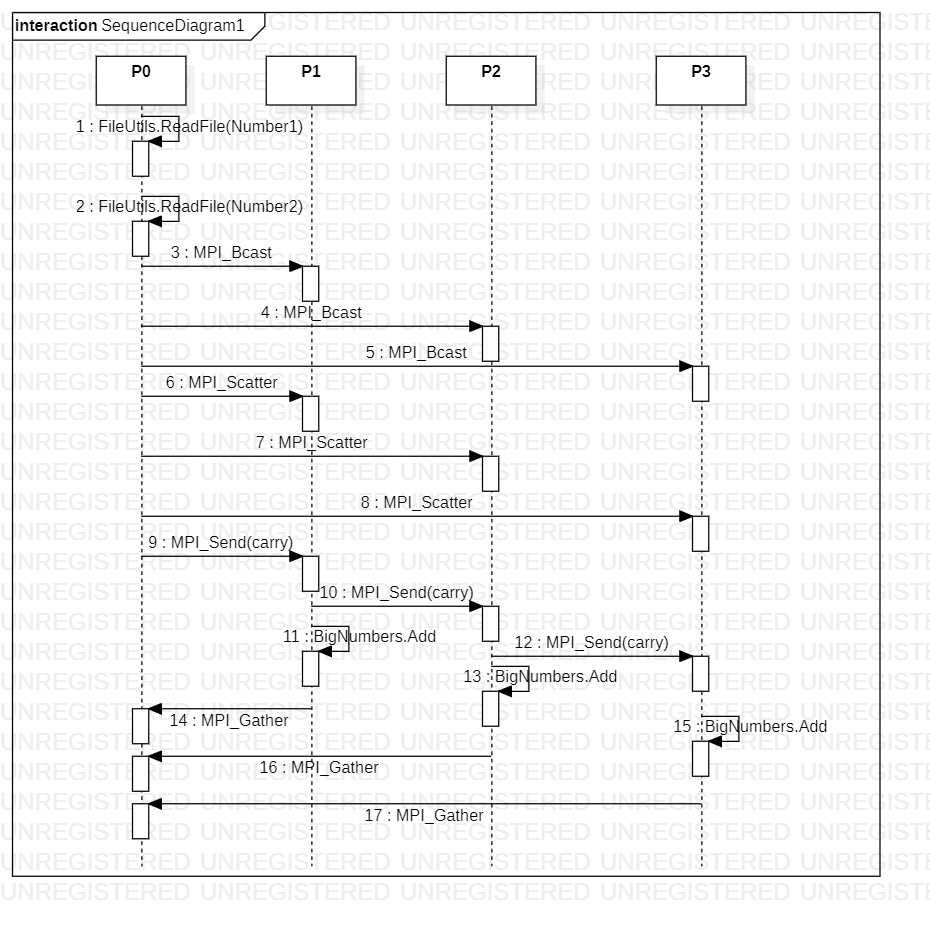
3) un process cu id<>0 primeste setul de cifre de la procesul 0 si face adunarea intr-un vector rezultat si actualizeaza “reportul”(carry”) pe care il trimite la procesul urmator (atentie un proces cu id (id<>1, id<>0) primeste informatie de la procesul 0 si de la procesul (id-1) dar ordinea intre cele 2 nu este sigura … se cere sa se foloseasca MPI\_Irecv )

4) rezultatul final se obtine in procesul 0 prin agregarea rezultatelor folosind transmitere asincrona 5) procesul 0 scrie rezultatul in fisierul “Numar3.txt”

Varianta 3 [Optional] [+ 2 puncte] - o varianta propusa de catre fiecare cu conditia sa fie diferita de cele 2 anterioare.

Comparati performantele obtinute folosind cele 3 variante. Teste:

1) Numar 1 = “123456789123456789” = Numar2 2) N\_1=1000 si N2=1000 3) N\_1=100 si N\_2=100000

Diagrama de secvență corespunzătoare variantei 1 este aceasta:

Observăm următoarele rezultate pe o configurație cu procesor Intel Core i5-8250U, 8 GB RAM DDR4 la 2400 MHz și Windows 10 (build 19041.572):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Run no. | Execution time | No. of processes | Transmission type | N1 | N2 |
| 1 | 0.0048 | 2 | isend | 1000 | 1000 |
| 2 | 0.006 | 4 | isend | 1000 | 1000 |
| 3 | 0.0068 | 6 | isend | 1000 | 1000 |
| 4 | 0.0088 | 8 | isend | 1000 | 1000 |
| 5 | 0.0154 | 16 | isend | 1000 | 1000 |
| 6 | 0.27 | 2 | isend | 10 | 100000 |
| 7 | 0.249 | 4 | isend | 10 | 100000 |
| 8 | 0.2534 | 6 | isend | 10 | 100000 |
| 9 | 0.2886 | 8 | isend | 10 | 100000 |
| 10 | 0.3194 | 16 | isend | 10 | 100000 |
| 11 | 0.0042 | 2 | scatter | 1000 | 1000 |
| 12 | 0.0056 | 4 | scatter | 1000 | 1000 |
| 13 | 0.006 | 6 | scatter | 1000 | 1000 |
| 14 | 0.007 | 8 | scatter | 1000 | 1000 |
| 15 | 0.012 | 16 | scatter | 1000 | 1000 |
| 16 | 0.2328 | 2 | scatter | 10 | 100000 |
| 17 | 0.2548 | 4 | scatter | 10 | 100000 |
| 18 | 0.2462 | 6 | scatter | 10 | 100000 |
| 19 | 0.2576 | 8 | scatter | 10 | 100000 |
| 20 | 0.2734 | 16 | scatter | 10 | 100000 |
| 21 | 0.0036 | 1 | sequential | 1000 | 1000 |
| 22 | 0.2436 | 1 | sequential | 10 | 100000 |
| 23 | 0.001 | 1 | sequential | 18 | 18 |
| 24 | 0.0022 | 2 | scatter | 18 | 18 |
| 25 | 0.0028 | 4 | scatter | 18 | 18 |
| 26 | 0.0046 | 8 | scatter | 18 | 18 |
| 27 | 0.0072 | 16 | scatter | 18 | 18 |
| 28 | 0.0014 | 2 | isend | 18 | 18 |
| 29 | 0.0028 | 4 | isend | 18 | 18 |
| 30 | 0.0054 | 8 | isend | 18 | 18 |
| 31 | 0.0096 | 16 | isend | 18 | 18 |

Graficul rulării programelor arată astfel:

Fig. 1 N1=10, N2=100000, program cu transmisie prin scatter

Fig. 2 N1=N2=18, program cu transmisie prin Isend asincron

Fig. 3 N1=10, N2=100000, program cu transmisie prin Isend asincron