

### Laborator L3

#### MPI

(deadline saptamana 7)

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 distribuie proceselor folosind MPI\_Scatter (daca nu este valabila conditia  $p|N$ , unde  $N=\max\{N_1, N_2\}$ ,  $N_1$  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) trimite “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 trimite procesului “id\_proces\_curent”
  - c. incrementeaza “id\_proces\_curent”
- 3) un process cu  $id \neq 0$  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  $id \neq 1$ ,  $id \neq 0$ ) 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  $N_2=1000$
- 3)  $N_1=100$  si  $N_2=100000$

Exemplu de adunarea a 2 numere mari (secvential):

	unitati	zeci	sute	mii	Zeci de mii	...
Nr1	3	6	9	2	...	...
Nr2	6	8	6	2	...	...
Carry	0	1	1	0	...	...
Rezultat	9	4	6	5	...	...

$$\text{Rezultat}[i] = \text{Nr1}[i] + \text{Nr2}[i] + \text{Carry}[i-1]$$

Pentru sute avem: Nr1 (9) + Nr2 (6) + carry de la zeci (1) = 16 (6 si carry 1)

Obs: Pentru adunare carry este intotdeauna egal cu unitatea (1).

Este posibil ca vectorul Rezultat sa fie mai mare cu un element (o cifra).

Exemplu pentru variantele paralele (Var 1 si Var 2):

