# Tema 3

Cerinta

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 estepe prima pozitie.Cele 2 numere mari se citesc din fisierele “Numar1.txt” (un numar cu N\_1 cifre) si “Numar2.txt”(un numar cu N\_2 cifre).Fiecare din aceste fisiere contine la inceput un numar (N) care reprezinta numarul de cifre si apoi cifrele numarului respectiv.

Functii

Proiectul are 4 functii

* generateNumber(digitsCount: Int, filename: String)
  + genereaza un numar cu digitsCount cifre si il scrie in fisier
* run1()
  + rezolva problema folosind Send si Recv
* run2()
  + rezolva problema folosinf Scatterv si Gatherv
* main()
  + punctul de intrare in program

Partitionarea datelor

Procesul de rang 0 citeste numerele din fisiere si de stocheaza in doua tablouri statice, primele cifre din tablou fiind cele mai putin semnificative. In cazul in care numerele nu au acelasi numar de cifre, numarul mai mic este completat cu 0. Apoi, datele sunt impartite in parti de lungime n/p +- 1, une n este dimensiunea celui mai mare numar, iar p – numarul de procese. Procesul master trimite fiecarui proces slave un start, end si partile corespunzatoare din numere.

Fiecare proces slave primeste de la root un start, end si partile din numere, dar si carry de la procesul anterior. Exceptie face procesul de rang 1, care primeste carry = 0 de la root. Fiecare proces calculeaza partea lui de numar, cu formula result[i] = first[i] + second[i] + carry, i = start, end. Apoi, procesul trimite la root start, end si result, dar si carry procesului de rang myRank + 1, mai putin procesul de rang p – 1 care trimite la root.

Cazuri de testare

P = 2

|  |  |  |
| --- | --- | --- |
| **Varianta** | **Caz** | **Timp** |
| **Varianta 1** | Numar1 = Numar2 = 123456789123456789 | 2.4727 |
| N1 = 1000, N2 = 1000 | 3.9179 |
| N1 = 100, N2 = 100000 | 56.1735 |
| **Varianta 2** | Numar1 = Numar2 = 123456789123456789 | 3.0447 |
| N1 = 1000, N2 = 1000 | 3.4267 |
| N1 = 100, N2 = 100000 | 53.9225 |

P = 4

|  |  |  |
| --- | --- | --- |
| **Varianta** | **Caz** | **Timp** |
| **Varianta 1** | Numar1 = Numar2 = 123456789123456789 | 4.7808 |
| N1 = 1000, N2 = 1000 | 5.8302 |
| N1 = 100, N2 = 100000 | 57.086 |
| **Varianta 2** | Numar1 = Numar2 = 123456789123456789 | 4.2278 |
| N1 = 1000, N2 = 1000 | 4.7976 |
| N1 = 100, N2 = 100000 | 62.4297 |

P = 8

|  |  |  |
| --- | --- | --- |
| **Varianta** | **Caz** | **Timp** |
| **Varianta 1** | Numar1 = Numar2 = 123456789123456789 | 7.6847 |
| N1 = 1000, N2 = 1000 | 11.5999 |
| N1 = 100, N2 = 100000 | 70.5702 |
| **Varianta 2** | Numar1 = Numar2 = 123456789123456789 | 10.387 |
| N1 = 1000, N2 = 1000 | 11.6073 |
| N1 = 100, N2 = 100000 | 72.6792 |

P = 16

|  |  |  |
| --- | --- | --- |
| **Varianta** | **Caz** | **Timp** |
| **Varianta 1** | Numar1 = Numar2 = 123456789123456789 | 16.5382 |
| N1 = 1000, N2 = 1000 | 19.2196 |
| N1 = 100, N2 = 100000 | 76.9239 |
| **Varianta 2** | Numar1 = Numar2 = 123456789123456789 | 17.3332 |
| N1 = 1000, N2 = 1000 | 18.5641 |
| N1 = 100, N2 = 100000 | 90.606 |

Se observa ca timpul de executie creste o data cu cresterea numarului de procese, iar varianta 1 este putin mai rapida fata de varianta 2, dar nu exista o imbunatatire substantiala.