Documentatie Lab2 PPD

Cerinte

a)Suma a 2 numere mari folosind MPI, procesul 0 citeste numerele si apoi le distribuie.

b)Suma a 2 numere mari folosind MPI, toate procesele sunt implicate in citire.

Descriere:

Varianta a) Procesul 0 citește ambele numere din fisier si le distribuie in mod egal fiecarui proces. Fiecare proces asteapta si va trimite la randul sau transportul efectuat in urma adunari in intervalul acordat.Procesele ce au primit un interval trimit procesului 0 rezultatul in urma adunarii, procesul 0 primeste si scrie in fisier rezultatul final.

Varianta b)Fiecare proces citeste un interval din fisier comform formulei stabilite anterior, ele primesc tranportul si il transmit mai departe procesului urmator, apoi trimit procesului 0 rezultat partial, procseul 0 primeste rezultatele partiale si afiseaza in fisier rezultatul final.

Pentru implementare am folosit libraria MPI in special functiile MPI\_Send si MPI\_Recv.

Proiectare C++

Functii folosite:

void computeV1(std::string file, int idProces, int nrProcese);

-compute the sum -first Version

File- the name of the file with the numbers

idProces -the current process

nrProces - the number of processes

void computeV2(std::string file, int idProces, int nrProcese);

-compute the sum -secondVersion

File- the name of the file with the numbers

idProces -the current process

nrProces - the number of processes

void write(std::string file, int \*\*a, int \*count, int row);

-write the result in the file

File- the name of the file

A the result

Count the length

Row -number of rows

int main(int argc, char\*\* argv);

Start the aplication

Testare

|  |  |  |  |
| --- | --- | --- | --- |
| Dimension | Nr procese | V1 | V2 |
| 10 | 2 | 10 | 8 |
| 10 | 4 | 9 | 8 |
| 10 | 6 | 11 | 11 |
| 10 | 8 | 11 | 11 |
| 100 | 2 | 7 | 9 |
| 100 | 4 | 8 | 10 |
| 100 | 6 | 9 | 9 |
| 100 | 8 | 15 | 12 |
| 1000 | 2 | 8 | 10 |
| 1000 | 4 | 7 | 10 |
| 1000 | 6 | 10 | 10 |
| 1000 | 8 | 13 | 11 |
| 10000 | 2 | 8 | 6 |
| 10000 | 4 | 12 | 7 |
| 10000 | 6 | 12 | 8 |
| 10000 | 8 | 13 | 9 |
| 100000 | 2 | 18 | 7 |
| 100000 | 4 | 17 | 8 |
| 100000 | 6 | 19 | 9 |
| 100000 | 8 | 25 | 15 |
| 1000000 | 2 | 109 | 8 |
| 1000000 | 4 | 111 | 10 |
| 1000000 | 6 | 113 | 12 |
| 1000000 | 8 | 110 | 16 |

Concluzia

Versiunea in care fiecare proces este implicat in citire este mult mai eficienta decat versiunea in care primul proces citeste intreg numarul si apoi il distribuie.