Bucșa Ecaterina, gr. 231

**Laborator 3**

**-documentație-**

Calculați suma a 2 numere mari (problema A. laborator 1) folosind MPI.

Cele două numere se citesc din două fișiere, iar rezultatul se salvează într-un alt fișier.

Se cere realizarea a două variante:

a) Procesul 0 citește numerele și apoi le distribuie.

b) Toate procesele sunt implicate în citire.

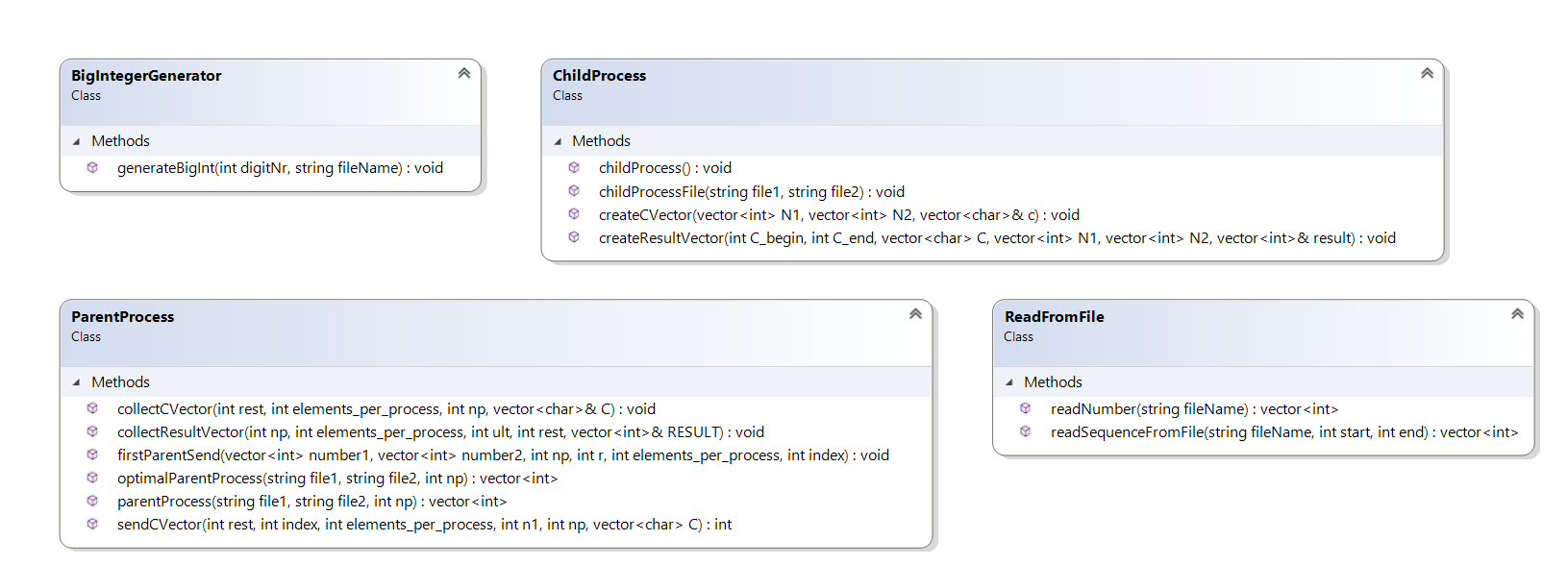
Constrângere: un proces trebuie să stocheze doar date pe care le procesează (cu execepția procesului 0 în cazul a)!

Analiza comparativă a performanței.

Ideea algoritmului utilizat: Procesul 0 citește cele 2 numere din fișiere și le distribuie în mod echitabil numărului de procese, trimițând mai întâi dimensiunea vectorilor pe care va trebui să îi adune apoi vectorii. Procesele copil adună cei doi vectori și formează vectorul C (C-carry, M-maybe, N- never). Fiecare proces copil trimite acest vector C părintelui care primește vectorii și îi pune în ordinea corespunzătoare ținând cont de faptul că ordinea de primtire nu este neapărat cea în care s-a trimis. După ce construiește acest vector îl trimite pe tot proceselor copil împreună cu un start și un end, reprezentând porțiunea care trebuie prelucrată de acel proces copil. Apoi procesele copil parcurg vectorul C (care ne spune dacă pe o anumită poziție avem transport), contruiește vectorul rezultat și în același timp adună și transportul. Vectorul rezultat este trimis procesului părinte care le primește și la fel cum a construit vectorul C, așa va construi și vectorul final al rezultatului.

Pentru varianta optimizată am procedat aproximativ la fel, doar că inițial procesul părinte nu trimite proceselor copil vectorii pe care trebuie să îi adune ci doar un start și un end reprezentând de unde până unde trebuie să prelucreze. Fiecare proces citește din fișier doar elementele de pe pozițiile din intervalul [start,end).

Diagrama de clase C++:



Cazuri de testare:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. cifre | Limbaj | Nr. procese | Varianta a | Vatianta b |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | C++ | 2 | 0.0128515 | 0.0137442 |
| 10 | C++ | 2 | 0.0122835 | 0.013952 |
| 10 | C++ | 4 | 0.013922 | 0.0147808 |
| 100 | C++ | 2 | 0.0137974 | 0.0139713 |
| 100 | C++ | 4 | 0.0139143 | 0.0163724 |
| 500 | C++ | 4 | 0.0151862 | 0.0154556 |
| 500 | C++ | 8 | 0.0133415 | 0.0184372 |
| 1000 | C++ | 4 | 0.0148307 | 0.0145402 |
| 1000 | C++ | 8 | 0.0140935 | 0.0177834 |
| 10000 | C++ | 4 | 0.0145135 | 0.0151209 |
| 10000 | C++ | 8 | 0.0153605 | 0.0171088 |
| 100000 | C++ | 2 | 0.0176498 | 0.0186251 |
| 100000 | C++ | 4 | 0.0174342 | 0.0238511 |
| 500000 | C++ | 2 | 0.0540512 | 0.03299 |
| 500000 | C++ | 8 | 0.039869 | 0.0248671 |
| 5000000 | C++ | 4 | 0.337897 | 0.211467 |
| 5000000 | C++ | 8 | 0.255778 | 0.128033 |
| 9000000 | C++ | 4 | 0.680519 | 0.356354 |
| 9000000 | C++ | 8 | 0.541585 | 0.244426 |