|  |  |  |
| --- | --- | --- |
| **Tip Matrice** | **Nr. procese** | **Timp de executie** |
| N=M=1000 | P = 5 | T1 = 3545.735  T2 = 1687.42 |
| P = 9 | T1 = 3568.945  T2 = 1692.941 |
| P = 21 | T1 = 3557.827  T2 = 1691.88 |

Observatii:

* Timpii de executie nu variaza prea tare in functie de nr. de procese, insa putem observa ca cel mai bun timp l-am obtinut pentru p=5. Putem observa si ca T1 este considerabil mai mare decat T2, ceea ce inseamna ca citirea si afisarea consuma mai mult timp, decat calculul efectiv al convolutiei.
* Timpii obtinuti sunt mult mai mari decat cei obtinuti la laboratorul 2, atat la metoda secventiala cat si cea multithreading (unde am obtinut aprox. 18-34 ms - multithreading, 28 ms - secv).

Detalii Implementare:

* procesul master transmite celorlalte procese matricea Kernel, N/(p-1) linii, si indicele de start si end al acestora. Citeste de la celelalte procese indicele liniilor si liniile calculate si scrie in fisier liniile preluate.
* pentru fiecare proces cu world\_rank-ul diferit de 0, preiau Kernel-ul, indicele de start si end al liniilor, apoi citesc liniile. Verific daca e primul sau ultimul proces, iar in functie de indicele acesta transmite prima lui linie la procesul anterior, ultima lui linie la procesul urmator, iar apoi citeste ultima linie a procesului anterior si prima linie a procesului urmator.
* dupa preluarea liniilor calculez convolutie pentru elementele de pe indicii de la start la end, apoi returnez la procesul master indicii linilor si liniile calculate.