Analiza- Lab3\_PPD

Varianta 1-Timpi

N=M=1000

|  |  |
| --- | --- |
| P=5 | 3400.1 ms |
| P=9 | 3397.38 ms |
| P=21 | 3497.01 ms |

Varianta 2-Timpi

N=M=1000

|  |  |
| --- | --- |
| P=4 | T1 8.95355  T2 3405.61735 |
| P=8 | T1 15.86725  T2 3459.7601 |
| P=20 | T1 27.3599  T2 3434.40385 |

Lab2-Timpi

N=M=1000

|  |  |
| --- | --- |
| Secvential | 190.694 |
| P=4 | 53.9644 |
| P=8 | 42.5136 |
| P=20 | 47.7283 |

Daca facem o comparative cu rezultatele din cadrul laboratorului 2 putem observa ca obtinem timpi mai buni daca folosim thread-uri, nu procese.

Putem observa ca numarul de procese nu influenteaza foarte mult timpii de executie, insa, pentru 4 sau 8 procese care prelucreaza datele obtinem timpi mai buni decat pentru 20 de procese care prelucreaza datele.

Urmarind datele din tabele putem observa ca timpii sunt foarte similari pentru cele doua variante(luam in calcul T1+T2), insa varianta 1, cea in care parintele(procesul 0) nu calculeaza noi valori este mai rapida.

Ca si implementare, varianta 1 mi se pare mai intuitiva, atat pentru faptul ca procesul parinte se ocupa de transmiterea/primirea datelor si nu prelucrarea acestora, cat si pentru ca se face distributia pentru procese de catre parinte si prin functii ajutatoare.