**Documentarea performanței**

**Java:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tip matrice** | **Nr threads** | **Timp execuție** | **Comanda** |
| N=M=10  n=m=3 | secvențial | 0,131834 | .\scriptJSequential.ps1 MainSequential "N=M=10 si n=m=3" 10 0 0 |
| 4 | 1,069432 | .\scriptJ.ps1 MainParallel "N=M=10 si n=m=3" 4 10 0 0 |
| N=M=1000  n=m=5 | secvențial | 37,091976 | .\scriptJSequential.ps1 MainSequential "N=M=1000 si n=m=5" 10 1 1 |
| 2 | 25,66447667 | .\scriptJ.ps1 MainParallel " N=M=1000 si n=m=5" 2 10 1 1 |
| 4 | 22,39992667 | .\scriptJ.ps1 MainParallel " N=M=1000 si n=m=5" 4 10 1 1 |
| 8 | 23,55493706 | .\scriptJ.ps1 MainParallel " N=M=1000 si n=m=5" 8 10 1 1 |
| 16 | 27,68397467 | .\scriptJ.ps1 MainParallel " N=M=1000 si n=m=5" 16 10 1 1 |
| N=10 M=10000  n=m=5 | secvențial | 8,417732 | .\scriptJSequential.ps1 MainSequential "N=10 M=10000 si n=m=5" 10 2 1 |
| 2 | 9,534022667 | .\scriptJ.ps1 MainParallel " N=10 M=10000 si n=m=5" 2 10 2 1 |
| 4 | 9,819471333 | .\scriptJ.ps1 MainParallel " N=10 M=10000 si n=m=5" 4 10 2 1 |
| 8 | 10,77548267 | .\scriptJ.ps1 MainParallel " N=10 M=10000 si n=m=5" 8 10 2 1 |
| 16 | 10,376898 | .\scriptJ.ps1 MainParallel " N=10 M=10000 si n=m=5" 16 10 2 1 |
| N=10000 M=10  n=m=5 | secvențial | 13,828236 | .\scriptJSequential.ps1 MainSequential " N=10000 M=10 si n=m=5" 10 3 1 |
| 2 | 14,900804 | .\scriptJ.ps1 MainParallel "N=10000 M=10 si n=m=5" 2 10 3 1 |
| 4 | 14,74691643 | .\scriptJ.ps1 MainParallel "N=10000 M=10 si n=m=5" 4 10 3 1 |
| 8 | 14,899276 | .\scriptJ.ps1 MainParallel "N=10000 M=10 si n=m=5" 8 10 3 1 |
| 16 | 14,17988733 | .\scriptJ.ps1 MainParallel "N=10000 M=10 si n=m=5" 16 10 3 1 |

**C++**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tip matrice** | **Tip alocare** | **Nr threads** | **Timp execuție** | **Comanda** |
| N=M=10  n=m=3 | static | secvențial |  |  |
| 4 |  |  |
| dinamic | secvențial |  |  |
| 4 |  |  |
| N=M=1000  n=m=5 | static | secvențial |  |  |
| 2 |  |  |
| 4 |  |  |
| 8 |  |  |
| 16 |  |  |
| dinamic | secvențial |  |  |
| 2 |  |  |
| 4 |  |  |
| 8 |  |  |
| 16 |  |  |
| N=10 M=10000  n=m=5 | static | secvențial |  |  |
| 2 |  |  |
| 4 |  |  |
| 8 |  |  |
| 16 |  |  |
| dinamic | secvențial |  |  |
| 2 |  |  |
| 4 |  |  |
| 8 |  |  |
| 16 |  |  |
| N=10000 M=10  n=m=5 | static | secvențial |  |  |
| 2 |  |  |
| 4 |  |  |
| 8 |  |  |
| 16 |  |  |
| dinamic | secvențial |  |  |
| 2 |  |  |
| 4 |  |  |
| 8 |  |  |
| 16 |  |  |

**Analiza**

Comparati performanta pentru fiecare caz – secvential versus paralel si variantele paralele intre ele.

Comparati timpii de executie obtinuti cu implementarea Java versus implementarea C++.

Comparati cele doua variante pentru implementarea C++.

Analiza trebuie evidentiata in documentatie.