|  |  |  |
| --- | --- | --- |
| Tip matrice | Nr. threads | Timp executie |
| N = M = 10  K = 3 | Secvential  4 pe coloane  4 pe linii | |  | | --- | | 0.01161 | | 0.45861 | | 0.59258 | |
| N = M = 1000  K = 5 | Secvential  1 pe coloane  2 pe coloane  4 pe coloane  8 pe coloane  16 pe coloane  1 pe linii  2 pe linii  4 pe linii  8 pe linii  16 pe linii | |  | | --- | | 16.27328 | | 8.64515 | | 15.25119 | | 18.89969 | | 24.29609 | | 12.45805 | | 11.70181 | | 19.04086 | | 26.98796 | | 33.54787 | | 34.3562 | |
| N = 10, M = 10000  K = 5 | Secvential  2 pe coloane  4 pe coloane  8 pe coloane  16 pe coloane  2 pe linii  4 pe linii  8 pe linii  16 pe linii | |  | | --- | | 0.65524 | | 1.11443 | | 1.41954 | | 2.20934 | | 3.10164 | | 1.59347 | | 1.9267 | | 2.87729 | | 4.35832 | |
| N = 10000, M = 10  K = 5 | Secvential  2 pe coloane  4 pe coloane  8 pe coloane  16 pe coloane  2 pe linii  4 pe linii  8 pe linii  16 pe linii | |  | | --- | | 0.67263 | |  | | 1.12902 | | 1.4949 | | 2.39692 | | 3.25434 | | 2.41937 | | 1.89491 | | 3.01127 | | 4.28494 | |

**JAVA**

**C++**

|  |  |  |  |
| --- | --- | --- | --- |
| Tip matrice | Nr. threads | | Timp executie |
| N = M = 10  K = 3 | Static pe linii 4  Dinamic pe linii 4  Static pe coloane 4  Dinamic pe coloane 4 | | 0.33328  0.32797  0.18238  0.39761 |
| N = M = 1000  K = 5 | Static pe linii | 1  2  4  8  16 | 11.12458  11.52698  13.25387  22.00431  41.52053  9.15951  9.20184  13.12715  17.47483  33.31838 |
| Dinamic pe linii | 1  2  4  8  16 |
|  | Static pe coloane | 1  2  4  8  16 | 11.67292  12.06742  14.64104  21.64237  41.4656 |
|  | Dinamic pe coloane | 1  2  4  8  16 | 9.51622  10.18036  11.20789  17.13228  30.76085 |
| N = 10, M = 10000  K = 5 | Static pe linii | 2  4  8  16 | 0.27812  0.30239  0.37453  0.56859 |
|  | Dinamic pe linii | 2  4  8  16 | 0.26978  0.3445  0.35001  0.60266 |
|  | Static pe coloane | 2  4  8  16 | 0.27537  0.27175  0.3872  0.59105 |
|  | Dinamic pe coloane | 2  4  8  16 | 0.21648  0.2656  0.38113  0.63602 |
| N = 10000, M = 10  K = 5 | Static pe linii | 2  4  8  16 | 0.34849  0.36577  0.4948  0.74306 |
|  | Dinamic pe linii | 2  4  8  16 | 0.3441  0.37058  0.46962  0.75344 |
|  | Static pe coloane | 2  4  8  16 | 0.27584  0.49265  0.5096  0.71813 |
|  | Dinamic pe coloane | 2  4  8  16 | 6,15615  3,82873  3,20303  2,71262  3,14204 |

Pentru cazurile cu matrici mai mici (N = M = 10, K = 3), în C++ timpul de execuție pentru variantele "Static pe coloane" și "Dinamic pe coloane" este semnificativ mai mic decât cel al variantelor "Static pe linii" și "Dinamic pe linii". Aceasta se datorează probabil modului în care datele sunt stocate în memorie și accesate în C++, ceea ce permite o mai bună localitate a datelor.

În cazurile cu matrici mai mari (N = M = 1000, K = 5), atât în Java cât și în C++, timpul de execuție crește odată cu creșterea numărului de fire de execuție. Cu toate acestea, la matricile mari, performanța C++ pare să fie mai bună decât cea a Java. Acest lucru se datorează probabil unei mai bune optimizări a memoriei și a limbajului C++ pentru operațiuni la nivelul limbajului de jos.

Cazurile cu matrici de dimensiuni diferite (N = 10, M = 10000 sau N = 10000, M = 10, K = 5) arată performanță similară în ambele limbaje. Cu toate acestea, în general, C++ pare să aibă o ușoară performanță superioară, ceea ce poate fi important pentru aplicațiile care necesită prelucrarea eficientă a unor volume mari de date.