# Laborator 2 – Documentație

De data aceasta, am folosit un vector dinamic / lista pentru retinerea rezultatelor fiecarui thread, ajungand astfel la o complexitate de spatiu de la O(N\*M) la O(N\*M/p).

## Astfel, datele obținute în urma rulării sunt următoarele:

1. **Pentru JAVA**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rulare JAVA | N | M | n | m | p | Time | Raport (parallel / sequencial) |
| seq | 10 | 10 | 3 | 3 | 2 | 807720 | - |
| par | 10 | 10 | 3 | 3 | 2 | 2279440 | 2.822067053 |
| par | 10 | 10 | 3 | 3 | 4 | 1994020 | 2.468702025 |
| seq | 1000 | 1000 | 5 | 5 | 2 | 13563600 | - |
| par | 1000 | 1000 | 5 | 5 | 2 | 14889100 | 1.097724793 |
| par | 1000 | 1000 | 5 | 5 | 4 | 14259240 | 1.051287269 |
| par | 1000 | 1000 | 5 | 5 | 8 | 16471880 | 1.214418001 |
| par | 1000 | 1000 | 5 | 5 | 16 | 15479300 | 1.141238314 |
| seq | 10 | 10000 | 5 | 5 | 2 | 37659240 | - |
| par | 10 | 10000 | 5 | 5 | 2 | 38449680 | 1.020989271 |
| par | 10 | 10000 | 5 | 5 | 4 | 39410300 | 1.046497486 |
| par | 10 | 10000 | 5 | 5 | 8 | 40487540 | 1.075102418 |
| par | 10 | 10000 | 5 | 5 | 16 | 42326180 | 1.123925496 |
| seq | 10000 | 10 | 5 | 5 | 2 | 1005540 | - |
| par | 10000 | 10 | 5 | 5 | 2 | 2091640 | 2.080116156 |
| par | 10000 | 10 | 5 | 5 | 4 | 2105920 | 2.094317481 |
| par | 10000 | 10 | 5 | 5 | 8 | 3202540 | 3.184895678 |
| par | 10000 | 10 | 5 | 5 | 16 | 3062800 | 3.045925572 |

OBS

Observăm o mică îmbunătățire a timpului de execuție față de laboratorul trecut, însă tot execuția secvențială domină, conform raportului.

1. **Pentru C++ (alocare statică)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rulare C++ Static | N | M | n | m | p | Time | Raport (parallel / sequencial) |
| S | 10 | 10 | 3 | 3 | - | 2274080 | 1 |
| P | 10 | 10 | 3 | 3 | 2 | 5091680 | 2.239006543 |
| P | 10 | 10 | 3 | 3 | 4 | 5204280 | 2.288521072 |
| S | 1000 | 1000 | 5 | 5 | - | 3278617240 | 1 |
| P | 1000 | 1000 | 5 | 5 | 2 | 3170805660 | 0.967116753 |
| P | 1000 | 1000 | 5 | 5 | 4 | 3048911120 | 0.92993811 |
| P | 1000 | 1000 | 5 | 5 | 8 | 3213448080 | 0.980122974 |
| P | 1000 | 1000 | 5 | 5 | 16 | 3093873300 | 0.943651873 |
| S | 10 | 10000 | 5 | 5 | - | 362473640 | 1 |
| P | 10 | 10000 | 5 | 5 | 2 | 327544660 | 0.903637186 |
| P | 10 | 10000 | 5 | 5 | 4 | 323993120 | 0.893839122 |
| P | 10 | 10000 | 5 | 5 | 8 | 328238260 | 0.905550704 |
| P | 10 | 10000 | 5 | 5 | 16 | 326682100 | 0.901257537 |
| S | 10000 | 10 | 5 | 5 | - | 335425600 | 1 |
| P | 10000 | 10 | 5 | 5 | 2 | 323078560 | 0.963189929 |
| P | 10000 | 10 | 5 | 5 | 4 | 313398000 | 0.934329401 |
| P | 10000 | 10 | 5 | 5 | 8 | 318569120 | 0.949745994 |
| P | 10000 | 10 | 5 | 5 | 16 | 318764100 | 0.950327286 |

OBS

Spre deosebire de laboratorul anterior, nu este o diferență atât de mare între timpul secvențial și cel paralel, însă mai mereu (cu excepția cazului în care N și M sunt 10), execuția paralelă este mai rapidă.

1. Pentru C++ (alocare dinamică)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rulare C++ Dinamic | N | M | n | m | p | Time | Raport (parallel / sequencial) |
| S | 10 | 10 | 3 | 3 | 2 | 2203560 | 1 |
| P | 10 | 10 | 3 | 3 | 2 | 4495720 | 2.040207664 |
| P | 10 | 10 | 3 | 3 | 4 | 4676780 | 2.122374703 |
| S | 1000 | 1000 | 5 | 5 | 2 | 3230078060 | 1 |
| P | 1000 | 1000 | 5 | 5 | 2 | 3099597320 | 0.959604462 |
| P | 1000 | 1000 | 5 | 5 | 4 | 2965829860 | 0.918191389 |
| P | 1000 | 1000 | 5 | 5 | 8 | 3095553120 | 0.958352418 |
| P | 1000 | 1000 | 5 | 5 | 16 | 3065105880 | 0.948926256 |
| S | 10000 | 10 | 5 | 5 | 2 | 335719900 | 1 |
| P | 10000 | 10 | 5 | 5 | 2 | 324609300 | 0.966905149 |
| P | 10000 | 10 | 5 | 5 | 4 | 315099360 | 0.938578142 |
| P | 10000 | 10 | 5 | 5 | 8 | 319146260 | 0.950632536 |
| P | 10000 | 10 | 5 | 5 | 16 | 326285660 | 0.971898478 |
| S | 10 | 10000 | 5 | 5 | 2 | 330741720 | 1 |
| P | 10 | 10000 | 5 | 5 | 2 | 315592020 | 0.954194772 |
| P | 10 | 10000 | 5 | 5 | 4 | 305517800 | 0.923735294 |
| P | 10 | 10000 | 5 | 5 | 8 | 310746620 | 0.939544669 |
| P | 10 | 10000 | 5 | 5 | 16 | 311667580 | 0.942329199 |

OBS

Identic ca la alocarea statică, diferența nu este foarte însemnată. Odată cu creșterea datelor, devine mai rapidă execuția paralelă.

1. **Rapoarte adiționale**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rulare | N | M | n | m | p | Raport JAVA / C++ Static | Raport C++ Static / C++ Dinamic |
| seq | 10 | 10 | 3 | 3 | 2 | 0.355185394 | 1.032002759 |
| par | 10 | 10 | 3 | 3 | 2 | 0.447679351 | 1.132561636 |
| par | 10 | 10 | 3 | 3 | 4 | 0.383150023 | 1.112791279 |
| seq | 1000 | 1000 | 5 | 5 | 2 | 0.004136988 | 1.015027247 |
| par | 1000 | 1000 | 5 | 5 | 2 | 0.004695684 | 1.022973416 |
| par | 1000 | 1000 | 5 | 5 | 4 | 0.00467683 | 1.028012821 |
| par | 1000 | 1000 | 5 | 5 | 8 | 0.005125921 | 1.038085265 |
| par | 1000 | 1000 | 5 | 5 | 16 | 0.005003211 | 1.009385457 |
| seq | 10 | 10000 | 5 | 5 | 2 | 0.103895114 | 1.079690659 |
| par | 10 | 10000 | 5 | 5 | 2 | 0.117387595 | 1.009042748 |
| par | 10 | 10000 | 5 | 5 | 4 | 0.121639311 | 1.028225256 |
| par | 10 | 10000 | 5 | 5 | 8 | 0.123348022 | 1.028488506 |
| par | 10 | 10000 | 5 | 5 | 16 | 0.129563818 | 1.001215009 |
| seq | 10000 | 10 | 5 | 5 | 2 | 0.002997803 | 1.014161745 |
| par | 10000 | 10 | 5 | 5 | 2 | 0.006474091 | 1.023722209 |
| par | 10000 | 10 | 5 | 5 | 4 | 0.006719634 | 1.025792933 |
| par | 10000 | 10 | 5 | 5 | 8 | 0.010052889 | 1.025173242 |
| par | 10000 | 10 | 5 | 5 | 16 | 0.009608359 | 1.022769516 |

OBS

Java este MULT mai rapid decât C++. Acest lucru se datoreaza în mare parte (în urma documentării) compilatorului Java (JIT), care se descurcă mult mai bine în multe cazuri de alocare / dealocare a memoriei, execuție paralelă, etc. De asemenea, este posibil ca laptopul meu să **urască multithreading-ul** (i did my best).

Varianta cu alocare dinamica din C++ este ușor mai lentă decât cea statică în fiecare caz (din nou, semn că **laptopul meu nu e chiar cooperativ**). Cel mai semnificativ este in cazul N=10, M=10, n, m = 3 cu 2 threaduri, raport = 1.133 (o diferență de 13.3%).