|  |  |  |
| --- | --- | --- |
| Nr Readers | Nr Workers | Timp executie |
| Secvential | | 19.6 |
| 1 | 4 | 18.7 |
| 1 | 6 | 18.7 |
| 1 | 8 | 18.8 |
| 1 | 16 | 18.9 |
| 2 | 4 | 18.7 |
| 2 | 6 | 19.1 |
| 2 | 8 | 18.6 |
| 2 | 16 | 18.9 |

*Timpi*

Se poate observa faptul ca in oricare caz paralele obtinem o imbunatatire fata de cazul secvential, chiar daca nu este o diferenta foarte mare, cazul cu 2 readeri si 8 workers fiind cel mai optim, cu o milisecunda mai rapid. Acesta lucru se datoreaza si specificatiilor laptopului, care ruleaza pe 14 core-uri, iar atunci cand rulam cazurile cu 16 workers, se si observa faptul ca ruleaza putin mai incet.

*Coada*

Pentru coada am folosit un semafor care semnaleaza cate element trebuiesc citite si am folosit un mutex pentru a bloca o metoda curenta care este apelata. Coada tine evidenta la numarul de readers pe care ii are. Cand se da push, se incearca sa se dea acquire, iar daca nu se reuseste sa da return la o valoare invalida. Cand de da pop, se incrementeaza semaforul ca a fost introdus un element.

*Lista*

In cazul listei, am folosit un mutex care blocheaza operatia de add cand se executa aceasta. Pentru adaugare, verificam daca nu a fost cineva care a incercat sa fraudeze, in caz afirmativ iesind din functie. Daca persoana de adaugat a incercat sa fraudeze, se adauga in lista persoanelor care au fraudat si se sterge din lista inlantuita. Altfel, verificam daca ID-ul exista in lista, in caz afirmativ ii crestem scorul si ii mutam locul in lista ca sa o mentinem sortata. In caz negativ, parcurgem lista si adaugam nodul in locul potrivit, fiind sortat dupa scor, iar daca scorul e egal, dupa ID.

*Implementare*

Un worker este blocat intr-un while pana cand nu ma exista readers sau date. Acesta tot incearca sa dea pop, iar daca primeste o valoare invalida, va fi pus in asteptare pentru 100 de milisecunde pentru a nu face overload la procesor. In cazul in care sunt mai multi readers, facem o impartire a fisierelor catre numarul acestora, fiecare primind numberOfCountries \* numberOfProblems / numberOfReaders.