|  |  |  |
| --- | --- | --- |
| **Nr. Threaduri Citire** | **Nr. Threaduri Workers** | **Timp de executie** |
| 1 | 4 | 798.189925 |
| 6 | 674.890266 |
| 8 | 651.1596 |
| 16 | 707.26167 |
| 2 | 4 | 108.9205 |
| 6 | 91.0054 |
| 8 | 97.8465 |
| 16 | 105.24587 |
| Secvential |  | 347.86934 |

Observatii:

* Metoda care foloseste 2 threaduri pentru citire este mult mai eficienta decat metoda secventiala
* Cel mai bun timp l-am obtinut folosind 2 threaduri pentru si citire si 8 workers
* Atunci cand am folosit un singur thread pentru citire, am obtinut cel mai ineficient timp

Implementare:

* Pentru metoda secventiala, pe parcurs ce citesc din fisier formez un nod din id-ul si punctajul participantului pe care apoi incerc sa il adaug in lista inlantuita in care retin clasamentul final. Prima data verific daca id-ul participantului este deja adaugat in lista. Apoi verific daca punctajul este egal cu -1, participantul va fi eliminat si voi adauga id-ul lui intr-o lista in care voi retine id-urile tuturor participantilor eliminati. In cazul in care punctajul este valid, daca participantul exista deja in lista, actualizam punctajul sau, altfel adaugam nodul in lista. La final, scriem lista in fisier.
* Pentru metoda paralela, impartim nr. de fisiere la threadurile care se vor ocupa cu citirea datelor. Initializam si pornim threadurile pt citire, apoi cele workers si asteptam finalizarea lor. Threadurile pt citire se ocupa de citire fisierelor cu indiciii de la start la end si adauga participantii intr-o coada, unde initial start = end = 0. Avem cat = nrFisiere/nrThreaduriCitire si rest = nrFisiere%nrThreaduriCitire. La fiecare pas end = start + cat, la care adaugam si 1 daca restul e mai mare decat 0. Coada va avea si un atribut care va retine nr de cititori activi, iar metodele ei vor fi sincronizate. Folosim si un mecanism wait notify: in metoda get care returneaza un nod din coada, daca ea este goala dar mai are cititori activi , asteapta, o notificare, care e trimisa in momentul in care se adauga un nod nou. Cand threadul de citire incheie parcurgerea fisierelor, decrementez nr. de scriitori ai cozi. Threadurile worker, parcurg coada, scot un element si blochez accesul concurent la lista, adaug nodul in lista, iar apoi deblocam accesul. Adaugarea in lista se face ca si in metoda secventiala.