**Analiză**

* Comparativ cu laboratorul anterior, unde întreaga listă era blocată pentru operații, implementarea actuală folosește un nivel de granularitate mai fin, la nivel de nod individual. Această schimbare îmbunătățește semnificativ performanța în mediul multithreading, eliminând mecanismele ineficiente de tip *busy-waiting*, care consumau inutil resursele CPU.

|  |  |  |
| --- | --- | --- |
| Nr. Readers | Nr. Workers | Timp Total Execuție |
| Secvențial |  | 344.3 |
| 4 | 2 | 40.02 |
| 4 | 4 | 26.98 |
| 4 | 12 | 19.16 |
| 2 | 2 | 52.65 |
| 2 | 4 | 27.81 |
| 2 | 12 | 20.22 |

**Implementare**

**Varianta Secvențială**

* Structura principală este o listă dublu înlănțuită, unde fiecare nod conține un obiect de tip Participant, format din trei câmpuri: ID, punctaj și țară.
* La inserarea unui participant:
  + Dacă ID-ul participantului există în lista participantiDescalificati, participantul este considerat descalificat și nu se efectuează nicio modificare.
  + Dacă participantul este deja în listă:
    - Dacă scorul este pozitiv, punctajul existent este actualizat și lista este resortată.
    - Dacă scorul este negativ, participantul este eliminat din listă.
  + Dacă participantul nu există în listă, acesta este adăugat și lista este sortată.
  + Orice participant cu punctaj negativ (-1) este tratat drept descalificat, fiind adăugat în vectorul participantiDescalificati și eliminat din listă, dacă era deja prezent.

**Varianta Paralelă**

* În varianta paralelă se adaugă o coadă thread-safe, protejată cu mutex-uri, care gestionează operațiile de inserare (push) și extragere (pop).
* Mecanismul include două variabile condiționale:
  + Inserarea (push) este permisă doar când în coadă sunt mai puțin de 50 de elemente.
  + Extragerea (pop) se poate realiza doar dacă coada nu este goală sau dacă mai există thread-uri de tip cititor active.
* La încheierea activității unui cititor, consumatorii sunt notificați.
* Lista utilizată în această implementare este una simplu înlănțuită, cu o santinelă ca nod inițial (cu ID = -1). Logica de funcționare:
  + Se menține vectorul de hash-uri pentru descalificare, accesul fiind protejat de un mutex. Mutex-ul este eliberat după blocarea mutex-ului asociat cu nodul cap al listei.
  + Lista este parcursă nod cu nod:
    - Dacă se găsește un nod cu același ID ca al participantului:
      * Pentru un scor pozitiv, punctajul participantului este actualizat.
      * Pentru un scor negativ, nodul este eliminat, iar legăturile listei sunt refăcute.
    - După procesarea fiecărui nod, mutex-ul asociat cu nodul anterior este eliberat, iar controlul trece la nodurile următoare.
  + Dacă nodul cu ID-ul participantului nu este găsit, iar participantul are un scor pozitiv, acesta este adăugat la finalul listei.
  + La final, mutex-ul ultimului nod procesat este eliberat. Dacă un nod a fost șters, memoria acestuia este eliberată.
* După încheierea procesării, lista este sortată și afișată, fără a include nodul santinelă.