**Analiza**

* Performanța programului secvențial este vădit una mai bună, fapt care se datorează blocărilor de tip excludere mutuală din implementarea multithreading, blocări care sunt la nivel de coadă și de listă, pentru fiecare operație (că vorbim despre adăugare de elemente în coadă, extragere, interogare de coadă vidă ori interogare de sfârșit citiri din fișiere, că vorbim despre adăugare în listă – care implică și sortarea listei ori ștergerea unui nod). Deci, conform datelor obținute, putem spune că programul secvențial este de cel puțin două ori mai rapid decât cel paralel.

|  |  |  |
| --- | --- | --- |
| **Nr readers** | **Nr workers** | **Timp executie total** |
| Secvential | cu timpul | 344.3 |
| 1 | 3 | 815.4 |
| 1 | 5 | 815.1 |
| 1 | 7 | 815.7 |
| 1 | 15 | 831 |
| 2 | 2 | 763.3 |
| 2 | 4 | 772.4 |
| 2 | 6 | 774.4 |
| 2 | 14 | 778.3 |

**Implementare**

* Secvențial
  + Definesc o structură de date care să simuleze o listă dublu înlănțuită, având ca dată un Participant (struct cu 3 proprietăți: id, punctaj și țară). La inserarea unui element în listă, se verifică hash-ul participantului. Dacă acesta a fost salvat într-un vector de hash-uri, atunci nu se mai face nimic (înseamnă că a fost eliminat din concurs acel concurent). Altfel, dacă a mai fost adăugat, i se incrementează scorul și se resortează lista, ori se adaugă pentru prima dată și se sortează lista. De fiecare dată când vom încerca să introducem un participant cu punctajul -1, acesta va fi marcat prin adăugarea hash-ului în vectorul amintit anterior și eliminarea (dacă acesta deja exista în listă) lui din listă.
* Paralel
  + Pe lângă structurile descrise mai sus, se adaugă încă una pentru implementarea unei cozi cu excluderi mutuale, atât la push, cât și la pop, respectiv la interogarea coadă vidă/ final citiri din fișiere. Structura pentru listă dublu înlănțuită se îmbogățește cu blocare mutuală peste adăugarea unui element în listă (blocare care se propagă, când sunt apelate, peste metodele de ștergere nod/ sortare listă). Avem modelul producător-consumator: producătorii adaugă în coadă participanți pe care îi citesc din fișiere, iar consumatorii, atât timp cât nu s-au terminat citirile realizare de producători și cât au ce consuma, scot participanți din coadă și îi adaugă în listă.