**Proiect Client-Server\_PPD 2025**

* Rusu Dana 236/1
* Stan Ariana-Maria 236/2

**Descriere generală:**

Acest sistem este compus dintr-un Server și Clienți. Fiecare Client reprezintă o țară și trimite rezultatele concurenților proprii către Server. Serverul calculează clasamentele pe baza datelor trimise, gestionează cererile de clasamente și trimite clasamentele finale către clienți sub formă de fișiere. De asemenea, serverul înregistrează toate acțiunile într-un fișier de log.

Clienții și serverul interacționează prin cereri HTTP. Fluxul dintre server și clienți presupune trimiterea datelor concurenților în blocuri, actualizarea clasamentelor și generarea rapoartelor.

**Fluxul dintre Client și Server:**

**1. Inițializarea Clientului:**

* Fiecare Client reprezintă o țară, iar fiecare țară trimite datele referitoare la concurenții săi către server.
* Clientul citește datele (ID-ul și scorul) dintr-un fișier și le grupează în blocuri de câte 20 de concurenți fiecare.
* Clientul trimite aceste blocuri de date serverului la intervale regulate (Δx), asigurându-se că fiecare bloc conține ID-urile și scorurile concurenților.

**2. Trimiterea datelor Concurenților:**

* Clientul trimite datele sub forma unui CompetitorBlock, care include:
  + **Numele țării**: Identifică țara.
  + **Concurenții**: Lista de perechi care conțin ID-ul concurentului și scorul acestuia.
* Clientul trimite fiecare bloc printr-o cerere HTTP POST către endpoint-ul /api/competitors.
  + Corpul cererii conține un obiect CompetitorBlock, serializat în format JSON.
* Serverul folosește p\_r threaduri de tip producer (ThreadPoolExecutor) pentru a procesa datele. Serverul primește acest bloc, îl procesează și adaugă datele concurenților în coada sa internă (MyQueue). Dacă un concurent are un scor de -1 (indicând plagiat), acesta este adăugat într-o listă neagră.

**3. Serverul procesează datele concurenților:**

* Serverul folosește p\_w threaduri de tip producer (ThreadPoolExecutor) pentru a procesa datele concurenților care sosesc asincron.
* Serverul adaugă datele în lista globală de clasamente (LinkedList), asigurându-se că informațiile fiecărui concurent sunt corect înregistrate, împreună cu numele țării sale.
* Când scorul este -1, concurentul este eliminat din clasamentul global și adăugat în lista neagră.
* Serverul are un număr fix de threaduri consumatoare (p\_w) care preiau datele din coadă și actualizează lista globală de clasamente.

**4. Calcularea Clasamentului:**

* Serverul calculează clasamentele pe baza listei globale de concurenți. Locul unei țări este determinat de suma scorurilor tuturor concurenților din acea țară.
* Clasamentele sunt actualizate periodic. Serverul actualizează clasamentul tarilor periodic (DELTA\_T). Actualizarea are loc doar dacă ultima actualizare a avut loc acum mai mult de DELTA\_T secunde.

**5. Cererea pentru Clasamentele Țărilor:**

* După trimiterea unei probleme, fiecare Client poate solicita clasamentul tarilor prin trimiterea unei cereri HTTP GET la /api/rankings/countries.
* Serverul verifică dacă clasamentele sunt actualizate (comparând timpul actual cu lastRankingUpdateTime). Dacă clasamentele sunt recente (adică ultima actualizare a fost acum mai puțin de DELTA\_T secunde), serverul trimite clasamentele salvate.
* Dacă clasamentele trebuie recalculate, serverul le calculează și apoi le trimite clientului.

**6. Cererea pentru Clasamentele Finale:**

* După trimiterea tuturor datelor, fiecare **Client** trimite o cerere pentru clasamentele finale folosind o cerere HTTP GET la /api/final-rankings.
* Serverul generează fișierele finale de clasamente (clasamentul global și clasamentul pe țări) și le trimite înapoi la client sub formă de byte arrays (encodate în base64). Aceste fișiere conțin clasamentele complete.
* Clientul decodează fișierele și le afișează pentru a vizualiza clasamentele, iar apoi își încheie activitatea.

**7. Încheierea Activității Serverului:**

* După ce Serverul a procesat toate cererile clienților și a generat fișierele finale de clasamente, se închide în mod corect.
* Serverul folosește un thread separat pentru a se închide după ce toate sarcinile au fost completate, asigurându-se că nu se termină prematur în timp ce procesează cereri.

**Fluxul detaliat pentru un singur Client:**

1. **Clientul pornește**:
   * Clientul se inițializează cu numele țării (de exemplu, C1, C2).
   * Clientul se conectează la server folosind un client HTTP și începe să trimită blocuri de date ale concurenților.
2. **Trimiterea datelor concurenților**:
   * Pentru fiecare bloc de date al concurenților, clientul serializa obiectul CompetitorBlock și îl trimite serverului printr-o cerere HTTP POST.
   * Clientul continuă să trimită blocuri până când toate datele sunt trimise.
3. **Cererea pentru clasamentele țărilor**:
   * După trimiterea unei probleme, clientul face o cerere pentru clasamentul pe tari prin trimiterea unei cereri HTTP GET la /api/rankings/countries.
   * Dacă serverul a calculat deja clasamentele în ultimele DELTA\_T secunde, serverul va trimite clasamentele; altfel, le va recalcula și le va trimite.
4. **Cererea pentru clasamentele finale**:
   * După trimiterea datelor pentru toți concurenții și primirea clasamentelor țărilor, clientul trimite cererea pentru clasamentele finale.
   * Serverul trimite cele două fișiere finale: clasamentul global și clasamentul pe țări. Clientul decodează fișierele și le afișează.
5. **Închiderea Clientului**:
   * După primirea clasamentelor finale, clientul închide activitatea.

**Fluxul pe partea serverului:**

1. **Recepționarea datelor concurenților**:
   * Serverul ascultă cererile POST pe /api/competitors și adaugă datele concurenților în coada sa.
   * Serverul procesează aceste date folosind threaduri producătoare-consumatoare pentru a actualiza lista globală de clasamente.
2. **Calcularea clasamentului**:
   * Serverul verifică periodic dacă trebuie să actualizeze clasamentele țărilor comparând timpul actual cu lastRankingUpdateTime.
   * Clasamentele sunt recalculat doar dacă ultima actualizare a avut loc acum mai mult de DELTA\_T secunde, altfel serverul trimite clasamentele salvate.
3. **Gestionarea cererilor de clasamente**:
   * Serverul gestionează cererile GET către /api/rankings/countries pentru a furniza clasamentele țărilor.
   * Dacă clasamentele sunt deja calculate recent, serverul trimite clasamentele; altfel, le recalculază și le trimite.
4. **Generarea clasamentelor finale**:
   * Serverul generează clasamentele finale și le salvează în fișiere (final\_global\_rankings.txt și final\_country\_rankings.txt).
   * Serverul trimite aceste fișiere clienților la cererea lor printr-o cerere GET la /api/final-rankings.

**Fluxul datelor:**

1. **De la Client la Server**:
   * Clienții trimit blocuri de date ale concurenților serverului folosind cereri POST la /api/competitors.
   * Clienții solicită clasamentele țărilor serverului folosind cereri GET la /api/rankings/countries.
   * Clienții solicită clasamentele finale serverului folosind cereri GET la /api/final-rankings.
2. **De la Server la Client**:
   * Serverul răspunde cererilor clienților pentru clasamentele țărilor cu clasamentele calculate.
   * Serverul răspunde cererilor clienților pentru clasamentele finale cu fișierele finale de clasamente.

**Concluzie:**

Acest sistem server-client gestionează trimiterea datelor concurenților, calcularea clasamentelor și generarea fișierelor finale de clasamente. Serverul procesează cererile asincron, iar clienții pot solicita rezultatele finale după completarea tuturor trimiterilor de date. Serverul procesează eficient cererile și răspunde cu clasamentele actualizate.

**Cazuri de testare:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Δx (sec)** | **p\_r** | **p\_w** | **Δt (ms)** | **Time (ms)** |
| 1 | 4 | 4 | 1 | 1056 |
| 2 | 1029 |
| 4 | 1035 |
| 2 | 1 | 2031 |
| 2 | 2011 |
| 4 | 2041 |
|  |  |  |  |  |
| 1 | 2 | 2 | 1 | 1038 |
| 2 | 1022 |
| 4 | 1043 |
| 2 | 1 | 2058 |
| 2 | 2030 |
| 3 | 2034 |
|  |  |  |  |  |
| 1 | 4 | 2 | 1 | 1020 |
| 2 | 1034 |
| 4 | 1045 |
| 2 | 1 | 2058 |
| 2 | 2032 |
| 4 | 2047 |
|  |  |  |  |  |
| 1 | 4 | 8 | 1 | 1034 |
| 2 | 1021 |
| 4 | 1045 |
| 2 | 1 | 2032 |
| 2 | 2043 |
| 4 | 2034 |
|  |  |  |  |  |