

Part2Part - Raport tehnic

Pasat Tudor Cosmin

Facultatea de informatica, UAIC, Iasi, Romania
<https://www.info.uaic.ro>

Abstract. In acest raport tehnic vom prezenta detalii de implementare al unui proiect de tip peer-to-peer, Part2Part. In sectiunile urmatoare vom prezenta tehnologiile utilizate, protocolul de comunicare dintre cele doua programe si o diagrama ce prezinta arhitectura aplicatiei.

1 Introducere

Part2Part este o aplicatie de tipul peer-to-peer pentru partajarea fisierelor. In acest raport am implementat o un sistem peer-to-peer hibrid, format dintr-un program server, ce poate fi rulat doar de catre autor, si un program client ce poate fi rulat de catre orice utilizator. Programul server este responsabil pentru primirea de catre cereri de la client, cautand fiecare fisier in baza de date pentru a afla ce utilizatori il au partajat, si transmite clientului detaliile de conectare ale acestor utilizatori. Dupa primirea acestor informatii, clientul se conecteaza la acel utilizator si ii transmite cererea de transfer.

2 Tehnologii utilizate

2.1 Comunicare

Comunicarea dintre server și client se face folosind un protocol TCP. Având în vedere că proiectul implementează un transfer de fișiere, calitatea informației transmise este mai importantă decât viteza transmiterii acesteia. De aceea nu puteam folosi un protocol UDP pentru comunicare. Pentru a putea primi mai multe cereri, serverul va crea pentru fiecare client un proces copil nou. La randul sau, clientul creeaza la inceput un proces copil responsabil cu primirea cererilor de transfer de fișiere, care creeaza pentru fiecare cerere un copil nou, ce isi termina executia la finalul transferului.

2.2 Baza de date

Serverul are propria baza de date, unde pastreaza detalii despre toti utilizatorii inregistrati, detaliile de conectare pentru fiecare client, ce sunt pastrate doar cat clientul este conectat la server, lista de fișiere partajate si o tabela ce contine ce fișiere are partajate fiecare utilizator. Clientul are la randul sau o baza de date unde se pastreaza path-ul fiecarui fisier partajat.

3 Arhitectura aplicației

Aplicatia este formata dintr-un program server, care indeplineste un set de functii in retea, si un numar nelimitat de clienti. Pentru a servi cererile in mod concurrent, serverul va crea un proces copil pentru fiecare client. In acelasi timp pentru trimiterea in paralel a mai multor fisiere, clientul creeaza pentru fiecare cerere un proces copil nou, ce isi termina executia la finalul transferului.

3.1 Diagrama

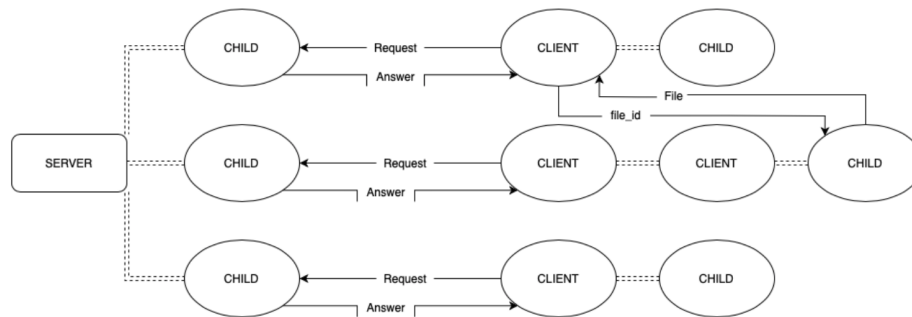


Fig. 1. Diagrama prezintă metoda de funcționare a aplicației în momentul în care sunt 3 utilizatori conectați

3.2 Baze de date

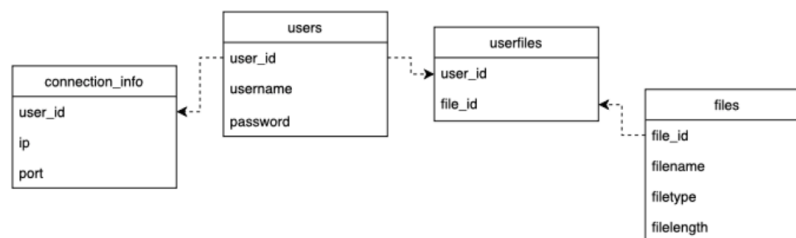


Fig. 2. Reprezentarea bazei de date din server

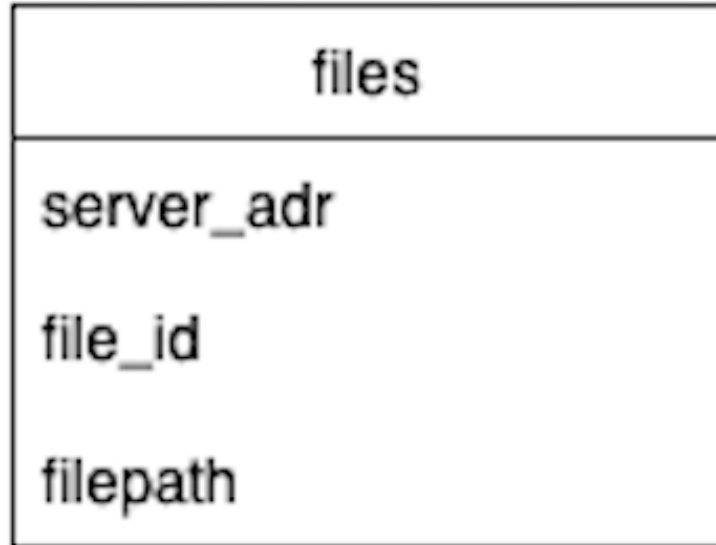


Fig. 3. Reprezentarea bazei de date din client

4 Detalii de implementare

Cele doua programe comunica intre ele pe baza unui protocol, ale carui specificatii le vom prezenta in aceasta sectiune. Comunicarea se face prin intermediul unor comenzi introduse de la tastatura in programul client, ce sunt delimitate de new line, si sunt interpretate de către server.

Comenzile ce pot fi introduse de catre client sunt:

- "signin" - permite utilizatorului sa se inregistreze, fiind identificat printr-un nume de utilizator si o parola. Acestea sunt pastrate in baza de date, la adaugare fiind generat un id unic pentru fiecare utilizator.
- "login" - dupa introducerea acestei comenzi, clientul trebuie sa isi scrie numele de utilizator si parola. Daca utilizatorul este acceptat, serverul va primi un port generat de client automat, si impreuna cu adresa si id-ul utilizatorului intr-o tabela din baza de date.
- "myfileslocal" - afiseaza toate fisierele partajate din baza de date locala, indiferent de utilizator.
- "myfilesserver" - afiseaza toate fisierele partajate ale utilizatorului din baza de date a serverului.
- "addfiles" - permite adaugarea unui fisier. Utilizatorul trebuie sa introduca path-ul fisierului, de unde va fi extras numele si tipul acestuia, cele doua

fiind transmise impreuna cu dimensiunea fisierului catre server. Un id specific fisierului este returnat si adaugat impreuna cu path-ul si adresa ip a serverului in baza de date a clientului.

- "removefile" - permite stergerea unui fisier in functie de id-ul dat de client.
- "searchfiles" - afiseaza toate fisierele din baza de date a serverului ce corespund criteriilor introduse:
 - "allfiles" - afiseaza toate fisierele.
 - "filename" - se va cere un sir de caractere si se returneaza toate fisierele ce au in nume acel sir.
 - "filetype" - toate fisierele de tipul dat.
 - "filelength" - toate fisierele de lungime data.
- "getfile" - utilizatorul poate introduce id-ul unui fisier, obtinut din comanda anterioara, si serverul va trimite, daca exista, detaliile de conectare al unui client ce are fisierul partajat. Daca apare o eroare la transfer, serverul va transmite datele urmatorului client, daca exista. Daca nu utilizatorul este notificat ca nu s-a putut face transferul.
- "quit" - aceasta comanda, dupa caz, va putea executa prima data comanda logout, dupa care incheie conexiunea cu serverul si inchide programul client.

5 Concluzii

Asta este tot ce am avut de spus despre implementarea unei aplicatii Part2Part. Pentru imbunatatirea proiectului putem realiza o interfata grafica pentru client, facand aplicatia mai placuta din punct de vedere estetic si mai usor de utilizat. In plus, putem modifica aplicatia, astfel incat informatiile tuturor clientilor ce au fisierul cerut partajat sa fie trimise utilizatorului, care va imparti in mod egal dimensiunea fisierului, astfel incat fiecare sa trimita o parte din el si transferul sa se faca mai repede pentru fisierele de dimensiune mai mare.

References

1. Pagina cursului: <https://profs.info.uaic.ro/~computernetworks/>
2. Pagina laboratorului: <https://profs.info.uaic.ro/~ioana.bogdan/>
3. SQLite <https://www.sqlite.org/index.html>
4. Stackoverflow: <https://stackoverflow.com>
5. Desenare diagrame: <https://app.diagrams.net>