

Tema 4 - SPRC

YET ANOTHER BITTORRENT!

Responsabil tema:

Florin Pop (florin.pop@cs.pub.ro)

Termen de predare:

Duminica, 22 Decembrie, 2013, ora 23.55

1 Specificatii functionale

Sa se implementeze o aplicatie peer-to-peer pentru download de fisiere din surse multiple, de tip BitTorrent (<http://www.bittorrent.com>).

Arhitectura aplicatiei consta intr-un nod central si orice numar de clienti ("peers"). Nodul central si fiecare client pot rula pe aceeasi masina virtuala sau pe masini virtuale diferite. Clientii pot intra in retea sau pot parasii reteaua in orice moment, fara o notificare prealabila (asincron).

Fisierele publicate rezida fizic pe clienti (fisierele nu vor fi incarcate pe nodul central). Fisierele publicate in cadrul retelei sunt unic identificate prin numele lor (cu alte cuvinte: doua fisiere cu acelasi nume, indiferent de locatia lor in retea, sunt identice bit cu bit). Cand un client publica un fisier, el il va imparti in fragmente de lungime egala, posibil cu exceptia ultimului fragment (care ar putea avea o lungime mai mica decat celelalte). Fiecare fragment astfel rezultat primeste un numar de ordine in secventa crescatoare, incepand cu 0. Fragmentarea unui fisier dat se face dupa un algoritm arbitrar (prin arbitrar se intelege la alegerea voastra), dar in mod identic pentru acelasi fisier, indiferent de clientul care publica fisierul si efectueaza fragmentarea, desi posibil in mod diferit de la fisier la fisier. Astfel, daca doi clienti publica un acelasi fisier (identificat prin numele sau), fragmentele acestuia pe cei doi clienti vor fi respectiv identice bit cu bit. Dupa fragmentarea initiala a unui fisier, clientul care il publica va anunta nodul central despre: (1) **numele fisierului**, (2) **dimensiunea totala a acestuia** si (3) **dimensiunea primului fragment** (restul informatiilor pot fi deduse).

Vom numi o structura de date care contine toate aceste informatii "**descrierea fisierului**". Pe langa descrierea fisierului, clientul care il publica va mai anunta nodul central despre adresa IP si portul pe care asculta cereri de download de la ceilalti clienti. Atat clientii cat si nodul central vor fi identificate prin adresa IP si portul pe care asculta. Adresa IP si portul nodului central se presupun cunoscute de toti clientii la initializare si nu se modifica in timpul rularii.

Nodul central va asocia IP-ul clientului si portul (presupuse fixe) cu descrierea fisierului si cu fiecare din numerele de secventa ale fragmentelor care compun fisierul. Ulterior fragmentarii initiale a unui fisier publicat, un client poate oferi spre download celorlalti clienti, la cererea acestora, orice fragment individual din fisier, identificat prin numarul sau de ordine.

Cand un client doreste sa downloadeze un fisier din retea, el va interoga mai intai nodul central, cu privire la fisierul identificat prin numele sau. Nodul central va raspunde acestei interogari cu:

- descrierea fisierului pentru fiecare fragment al fisierului (identificat prin numarul sau de ordine);
- o lista de clienti (identificati prin adresele IP si porturile lor) care au anuntat anterior ca detin respectivul fragment.

Clientul isi poate alege apoi clientii (peers) de la care sa descarce, eventual in paralel, diferitele fragmente individuale ale fisierului. Algoritmul de alegere a unui anumit client (pentru download) din lista asociata unui fragment de fisier nu este impus. Clientii pot cere doar fragmente individuale de fisiere de la alti clienti; daca se doreste transferul integral al unui fisier (chiar si de la acelasi client), se vor initia cereri separate pentru fiecare din fragmentele fisierului.

Imediat dupa downloadul unui fragment individual al unui fisier, clientul care l-a downloadat va anunta nodul central despre:

- numele fisierului;
- numarul de secventa al fragmentului.

In acest fel nodul central va putea include IP-ul acestui client pe lista asociata respectivului fragment al fisierului, iar alti clienti vor putea cere de la acest client respectivul fragment de fisier.

Pentru simplificare, vom presupune ca un fragment de fisier, o data publicat de un client, nu mai poate fi sters (unpublished) si ca, atata vreme cat este functional si accesibil, clientul va furniza celorlalti clienti respectivul fragment de fisier, la cererea lor. (Cu toate acestea, este posibil ca un client sa devina, temporar sau permanent, nefunctional sau inaccesibil prin retea, iar potentialii clienti care ar incerca sa interactioneze cu el vor fi toleranti la acest tip de erori).

Atat clientii cat si nodul central vor fi capabile sa serveasca mai multi clienti simultan.

2 Specificatii non-functionale

- Aplicatia trebuie sa fie scalabila cu numarul de clienti (peers) din retea si cu numarul total de fisiere publicate de acestia. Desi nu avem cerinte stricte legate de timing si alti parametri, va trebui sa aveti totusi in vedere aspectele legate de performanta si utilizarea resurselor.
- Erorile de comunicatie (si erorile in general) trebuie rezolvate "gracefully". In nici un caz nu sunt admise exceptii aruncate la runtime si netratate.
- Aplicatia trebuie sa fie corect sincronizata, fara alte ipoteze simplificatoare.

3 Ipoteze initiale

- IP-ul si portul nodului central sunt cunoscute de la inceput de toti clientii.
- IP-ul si portul fiecarui client se presupun fixe pe parcursul executiei aplicatiei (portul poate diferi de la client la client).
- Daca downloadul unui fragment de fisier este intrerupt (de exemplu, clientul de la care se downloadeaza devine inaccesibil), fragmentul partial posibil salvat pana in acel moment se ignora si nu va fi publicat.
- Clientul poate continua sa ofere spre download fragmente individuale de fisier, chiar daca nu a reusit sa downloadeze toate fragmentele fisierului.
- Nu ne intereseaza aspectele de autentificare (certificate, conexiuni securizate etc.) ale clientilor si fisierelor. Cu alte cuvinte, se presupune ca nu exista clienti malitiosi in retea.

4 Constrangeri de implementare

- Implementarea se va realiza folosind Java (Version 7).
- Atat clientii cat si nodul central sunt multithreaded.
- Pentru comunicatia peste retea, se vor prefera clasele din pachetul java.nio (clasa `SocketChannel` va fi preferata clasei `Socket`, etc.).
- Socketii neblocați vor fi preferati socketilor blocați.
- Pentru a facilita testarea, clasa client se va numi "Client" si va trebui sa extinda clasa `AbstractClient` (pusa la dispozitie).
- Pentru logging, se va folosi API-ul `log4j`. Logurile generate de aplicatia trimisa vor contine strict informatii legate de operatiile efectuate de clienti si de nodul central (formatul exact nu este specificat) si de eventualele erori (nu si mesaje de debug).
- Pentru lansarea in executie a serverului se va folosi urmatoarea comanda (`server_port` este portul pe care asculta serverul):

```
java Server server_port
```

- Pentru lansarea in executie a clientului se va folosi urmatoarea comanda (`client_port` este portul pe care asculta clientul, `server` IP-ul serverului si `server_port` portul pe care asculta serverul):

```
java Client server server_port client_port
```

5 Specificatii de deployment

Structura de directoare a temei trebuie sa includa cel putin urmatoarele:

- `src` : directorul radacina al surselor claselor de baza si al celor de test
- `classes` : directorul radacina al claselor compilate
- `lib` - jar-urile folosite de aplicatie, inclusiv `log4j<...>.jar`.
- `logs` : directorul cu loguri generate de aplicatie
- `doc` : documentatia temei; implementarea si utilizarea aplicatiei precum si descrierea algoritmilor folositi, cu accent pe portiunile de sincronizare intre threaduri.

Deployment-ul, compilarea si rulara aplicatiei se vor face folosind **ant**. Fisierul **build.xml** trebuie sa contina cel putin urmatoarele targeturi: **clean**, **build**, **run**. Tema trebuie organizata in asa fel incat deploymentul sa fie posibil pe orice masina (cu alte cuvinte, sa fie de tipul “extract-compile-and-run”, fara alte operatii de executat manual).

6 Observatii

- Inainte de trimiterea temei de casa trebuie sa cititi cu atentie regulamentul cursului de SPRC disponibil pe site-ul cursului si sectiunile 4 si 5.
- Data fiind complexitatea temei, va sugeram sa va alocati timp suficient pentru ea.
- Intrebarile cu privire la rezolvarea temei de casa se vor pune pe forumul dedicat acesteia, forum disponibil pe site-ul cursului. NU publicati bucati de cod sursa pe forum!