Sistem distribuit de sincronizare a fișierelor

Andrei Chelariu

Rezumat

Lucrarea de față are ca scop sincronizarea modificărilor care apar în cadrul unei ierarhii de directoare între mai multe noduri din sistem. Astfel, modificarea unui fișier pe unul din noduri va duce la replicarea modificării pe celelalte noduri din rețea. În vederea realizării aplicației, aceasta a fost separata în trei componente de sine stătătoare.

Componenta Passive Front End este un proces care detectează schimbările care apar în cadrul sistemului de fișiere și care le transmite sub forma de mesaje JSON către orice proces interesat. Aplicația utilizează serviciul Inotify, pus la dispoziție de nucleul sistemului de operare linux, pentru detectarea modificărilor și păstrează în memorie o replică a ierarhiei de directoare supravegheată. Provocările care au apărut în cadrul implementării acestei componente au constat în ajustarea replicii din memorie la modificările care apar pe disc.

Componenta Broadcast Mechanism este o bibliotecă dinamică ce oferă posibilitatea unui nod din sistem sa transmită și sa primească mesaje de la celelalte noduri din rețea. Componenta a fost realizată în C++ pentru un control mai bun al performanței, dar prezintă și o interfață către limbajul de programare lua. Provocările care au apărut în cadrul implementării acestei componente au constat în satisfacerea constrângerilor pe care trebuie sa le îndeplinească un sistem distribuit: atomicitate, consistență, izolare.

Componenta Replication Manager este un proces ce primește evenimentele de la Passive Front End, le filtrează și apoi le transmite către celelalte noduri din rețea folosind componenta Broadcast Mechanism. Componenta a fost scrisă în limbajul de programare lua pentru o mai mare flexibilitate și permite utilizatorului sa își definească propria funcție de rezolvare a conflictelor. Provocările care au apărut în cadrul implementării acestei componente au constat în rezolvarea conflictelor dintre modificările făcute local de utilizator și cele făcute de ceilalți utilizatori din rețea.

În concluzie, aplicația separa funcționalitate de baza în procese de sine stătătoare pentru o mai buna izolare în caz de erori și oferă utilizatorului posibilitatea de a modifica o parte din comportamentul acesteia prin intermediul definirii propriilor funcții de tratare a conflictelor.