

File Sharing Cloud

Dokumentacija

Uvod

Distribuirani sistem koji obezbedjuje rad sa ASCII kodiranim tekstualnim datotekama. Sistem korisniku omogućava sledeće:

- Dodavanje nove datoteke sa jedinstvenim nazivom i putanjom u sistem.
- Dohvatanje proizvoljne datoteke iz distribuiranog sistema.
- Brisanje datoteke, na lokalnom serventu.
- Topološka organizacija sistema omogućava brže dohvaćanje datoteka.
- Otpornost na otkaze.

Identifikacija cvora u mrezi

Svaki cvor u mrezi kod sebe I u porukama u cijem slanju I primanju ucestvuje, sadrzi objekat klase ServentInfo pomocu koga se identifikuje.

Ova klasa pruza sledece informacije:

- Id cvora
- Ip adresu cvora
- Port cvora
- Ko su mu susedi

BootstrapServer

Bootstrap je komponenta za koju ceo sistem podrazumeva da je uvek ukljućena, I ima za zadatak da zabeleži prijavljivanje ćvorova u mrežu I odjavljivanje, kao I da nasumicno izabere jedan od prijavljenih ćvorova kao odgovor ćvoru koji se prijavljuje.

U slićaju da je lista prijavljenih ćvorova prazna u trenutku kada se prijavljuje novi ćvor, to znaći da je taj ćvor prvi u sistemu I bootstrap mu odmah salje poruku dobrodoslice.

Komunikacija sa BootstrapServerom zapocinje kada pokrenemo novog Serventa i tada se on preko Socket-a javlja Bootstrap-u da ga ukljuci u sistem. Ukoliko je u tom trenutku broj cvorova u sistemu jednak nuli, to znaci da je upravo pokrenuti cvor prvi u sistemu i tada mu Bootstrap salje poruku **WelcomeMessage**, cime ga ukljucuje u sistem. U suprotnom izabrace proizvoljan port nekog od cvorova u sistemu i vratile mu na Socket, kako bi dalje Servent mogao njemu da se obrati za ukljucenje u sistem.

Poruke

Sve poruke koje se koriste u komunikaciji obavezno sadrže:

- Id cvora koji salje poruku
- Id cvora koji prima poruku
- Tip poruke

Ukljucivanje novog cvora u sistem

U slucaju da vec postoje aktivni cvorovi koji su deo sistema, tada nas Servent koji je upravo pokrenut dobija preko Socket-a informaciju o nasumicnom cvoru iz sistema i u tom trenutku salje **HiMessage** gde stavlja svoj port kao posiljalac, port dobijen od bootstrapa kao primalac i prosledjuje mu informacije o sebi. Taj cvor kada primi **HiMessage** salje nasem Serventu **WelcomeMessage**, a ostalim cvorovima u sistemu **NewNodeUpdateMessage** (broadcast tip poruke) kako bi i oni znali da se novi cvor ukljucio u sistem, tj kako bi azurirali svoje stanje.

Iskljucivanje cvora iz sistema

Kada se ServentMain proces prekine, tada se pokrece *stop* metoda koja kreira broadcast tip poruke **DeletedNodeUpdateMessage** koji obavestava ostatak sistema da je jedan cvor upravo ugasen.

Provera otkaza

PingMessage - svaki cvor salje svom buddy-u kako bi proverio da li je i dalje aktivan

PongMessage - odgovor na PingMessage

BackupMessage - na svaku desetu PingMessage salje se BackupMessage istom tom cvoru, koja u sebi nosi informacije o fajlovima koji se nalaze na nasem cvoru i sluzi kako bismo se osigurali u slucaju neplaniranog otkazivanja naseg cvora

Ukoliko posiljalac Ping poruke ne dobije odgovor u okviru slabe granice otkaza, tada salje novu Ping poruku sa oznakom "IsOk?", i to je salje daljem susedu, da on pokusa da kontaktira odsutni cvor, a on ako ne dobije odgovor opet u okviru slabe granice, javlja nam da je cvor otkazao i mi obavestavamo ostatak sistema da azurira stanje.

Zahtevanje kriticne sekcije

WantToken - saljemo iz naseg cvora ostatku sistema kada zatrazimo *lock()* kako bismo se upisali u red cekanja za Token

Token - salje se od cvora koji ga trenutno poseduje (kada se pozove *unlock()*) do cvora koji je sledeci u redu cekanja, omogucava ulazak u kriticnu sekciju

Fajl sistem

PullAskMessage

- trazimo od cvorova u sistemu da nam dobave fajl koji je korisnik zatražio, jer ga nemamo kod sebe
- Cvorovi koji nemaju taj fajl, prosledjuju ovu poruku dalje, kako bi se pretraga nastavila
- Onaj cvor koji ga ima ce ga vratiti putem Tell poruke

PullTellMessage

- Kada dobijemo ovu poruku ispisujemo sadrzaj fajla na konzolu