

RafBook

Distribuirani sistem za upravljanje fajlovima

Uvod

RafBook je distribuirani sistem za upravljanje fajlovima baziran na Chord (DHT) arhitekturi i Ricart-Agrawala algoritmu za međusobno isključivanje. Ricart-Agrawala algoritam je distribuirani algoritam za međusobno isključivanje koji osigurava pravedan pristup kritičnoj sekciji. Svaki čvor šalje zahtev svim ostalim čvorovima i čeka da dobije odgovore pre nego što uđe u kritičnu sekciju. Ovo obezbeđuje da samo jedan čvor u mreži može istovremeno pristupiti kritičnoj sekciji. Sistem omogućava efikasan raspored i pristup fajlovima u mreži sa više čvorova, obezbeđujući visoku dostupnost i otpornost na greške. Osnovni cilj sistema je omogućiti bezbedan i efikasan pristup kritičnim sekcijama u distribuiranom okruženju..

Osnovne karakteristike

- **Distribuirano skladištenje:** Fajlovi su raspoređeni na više čvorova unutar mreže koristeći Chord DHT protokol, što omogućava balansiranje opterećenja i otpornost na otkaze pojedinačnih čvorova.
- **Replikacija fajlova:** Sistem implementira mehanizme za replikaciju fajlova kako bi osigurao da su fajlovi dostupni čak i u slučaju kvara čvorova.
- **Brza pretraga:** Koristeći Chord algoritam, sistem omogućava efikasnu pretragu i pristup fajlovima putem ključnih vrednosti.

Komande

- **add_friend [adresa:port]** - Komanda koja prosleđeni port dodaje u listu prijatelja čvora koji pozove komandu.
- **add_file [filePath visibility]** – Komanda koja dodaje fajl sa određenom putanjom u sistem, sa private ili public vidljivošću. Vidljivost naglašava dostupnost fajla

ostalim čvorovima, private fajlovi su vidljivi samo prijateljima čvora koji je fajl dodao, dok su public fajlovi vidljivi svim čvorovima u sistemu.

- **view_file [filename]** - Komanda koja čvoru koji je pozove vraća zatraženi fajl u zavisnosti od dostupnosti, to jest. vidljivosti fajla.
- **remove_file [filename]** - Komanda koja briše fajl sa prosleđenim imenom.
- **stop** – Komanda koja gasi sve čvorove koji rade.

Opis sistema

Konfiguracija sistema

Konfiguracija sistema je realizovana kroz **properties** fajl koji sadrži ključne parametre kao što su broj servenata, maksimalan broj fajlova, port bootstrap komponente, slaba i jaka granica otkaza, kao i portovi čvorova u sistemu. Ova konfiguracija služi kao jedinstveni izvor istine za sistem i osigurava konzistentnost i efikasno upravljanje parametrima.

Bootstrap

Bootstrap server je ključna komponenta koja dodeljuje čvorovima odgovorne za dodavanje novih čvorova u sistem i njihovo isključivanje. Prilikom dodavanja novog čvora, bootstrap server dodeljuje nasumičan čvor kojem će se novi čvor javiti radi dodavanja u sistem. Ovaj proces je detaljno opisan u sledećem poglavlju.

Pokretanje sistema

Prilikom pokretanja sistema, čvorovi se javljaju bootstrap serveru koji im dodeljuje nasumičan čvor kojem će se obratiti za dodavanje u sistem. Ako je u pitanju prvi čvor, on se automatski dodaje. Čvor šalje **NEW_NODE** poruku čvoru koji ga treba dodati, a ta poruka se obrađuje u **NewNodeHandler** klasi. Ako čvor nije prethodnik, poruka se prosleđuje dalje dok ne dođe do čvora koji će ga dodati. Kada čvor primi **WELCOME** poruku, on se inicijalizuje i šalje **UPDATE** poruku svim čvorovima kako bi oni ažurirali svoje mape.

Dodavanje prijatelja

Kada čvor dodaje prijateljskog čvora, on prvo dodaje taj čvor u listu prijatelja i šalje mu obaveštenje. Drugi čvor, kada primi poruku, dodaje prvi čvor u svoju listu prijatelja, čime se uspostavlja dvosmerno prijateljstvo.

Dodavanje fajla

Dodavanje fajla se vrši komandom `add_file` koja zahteva putanju do fajla i vidljivost (privatna ili javna). `AddFileCommand` proverava da li je trenutni čvor odgovoran za čuvanje fajla. Ako jeste, fajl se dodaje u njegovu mapu vrednosti, a ako nije, `AddFile` poruka se prosleđuje dalje dok ne stigne do čvora koji je odgovoran za taj fajl.

Dohvatanje i čitanje fajla

Kada jedan čvor želi da pregleda fajl on šalje `FileAskMessage` poruku traženom čvoru. Traženi čvor odgovara sa `FileTellMessage` porukom. Nakon toga, čvor koji je inicirao zahtev može pristupiti tom fajlu ako ima dozvolu. Ako su fajlovi privatni, proverava se da li je čvor koji zahteva fajlove prijatelj sa vlasnikom fajlova pre nego što mu se omogući pristup.

Brisanje fajla

Brisanje fajla se obavlja komandom `remove_file`. Čvor koji je odgovoran za čuvanje fajla briše ga iz svoje mape vrednosti. Ako čvor nije odgovoran, šalje `FileRemoveMessage` poruku dalje dok ne nađe čvor koji je odgovoran za taj fajl. Kada čvor koji je odgovoran za fajl primi poruku, on briše fajl i šalje `FileRemovedMessage` poruku nazad čvoru koji je zahtevao brisanje kako bi ga obavestio o uspešnom brisanju.

Otpornost na otkaze

Sistem koristi Ricart-Agrawala algoritam za otkrivanje i oporavak od otkaza. Svaki čvor redovno šalje `TICK` poruke svom prethodniku i sledbeniku. Ako čvor ne odgovori na ove poruke, šalje se `SOFT_FAIL` poruka, a ako ni tada ne dođe do odgovora, šalje se `HARD_FAIL` poruka koja pokreće proces reorganizacije sistema. Ovaj proces uključuje ažuriranje strukture sistema kako bi se osiguralo da svi fajlovi ostanu dostupni.

Backup

Sistem redovno pravi rezervne kopije podataka. Svaki čvor šalje spisak svojih fajlova naslednicima iz tabele sukcesora, koji ih čuvaju kao rezervne kopije. Ove kopije omogućavaju oporavak podataka u slučaju otkaza čvora koji je bio odgovoran za originalne fajlove.

Heartbeat

Heartbeat mehanizam se koristi za detekciju otkaza čvorova. Sistem definiše slabe i jake granice otkaza. Ako čvor ne odgovori na **TICK** poruke, pokreće se proces reorganizacije sistema. Prilikom otkaza prethodnika, novi prethodnik preuzima odgovornost za fajlove i sistem se reorganizuje kako bi osigurao kontinuitet rada.

Luka Mitrović RN61/2020