**FILE SHARING CLOUD**

Dokumentacija projekta rađenog u okviru kursa

Konkurentni i distribuirani sistemi,

Računarski fakultet u Beogradu

Jovan Lazić RN 114/20

Uvod………………………………………………………………………………………………….2

Čvor…………………………………………………………………………………………………..2

Bootstrap čvor………………………………………………………………………………………..2

Arhitektura mreže…………………………………………………………………………………….2

Konkurentnost………………………………………………………………………………………..2

Konstante……………………………………………………………………………………………..3

Komande čvorova…………………………………………………………………………………….3

Otkaz čvorova………………………………………………………………………………………...3

Backup funkcionalnost……………………………………………………………………………….3

Detektovanja otkaza………………………………………………………………………………….3

Startovanje Bootstrap čvora………………………………………………………………………….3

Startovanje čvora i ulazak u mrežu…………………………………………………………………..3

Komunikacija………………………………………………………………………………………...5

Uvod

Ovaj program predstavlja distribuirani sistem koji konkurentno raspolaže sa tekstualnim fajlovima.

U ovom dokumentu je opisan način funkcionisanja sistema, protokol razmene poruke i sama rešenja nekih problema sa kojim se susreće naš sistem.

Čvor

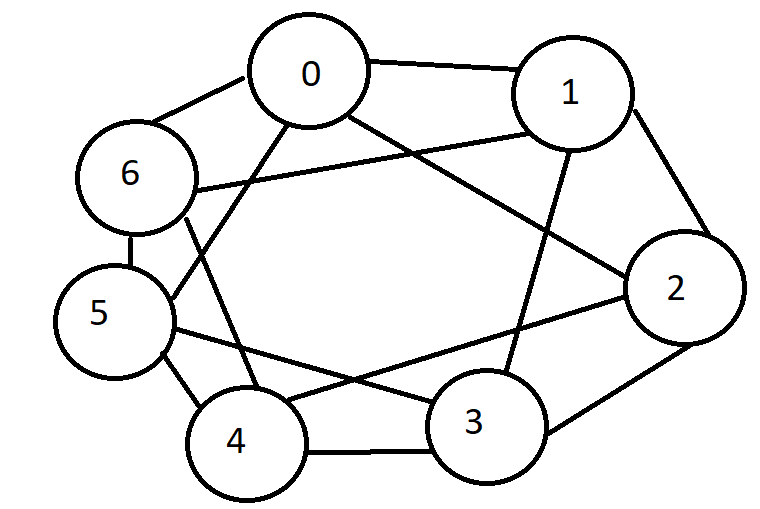
Node ili čvor jeste čvor koji ima svoj ID i služi kako bi slao i primao poruke sa određene IP adrese i porta. Čvor može da čuva fajlove, da ih povlači sa drugih čvorova, briše i čuva backup-ove drugih čvorova.

Bootstrap čvor

Bootstrap čvor jeste čvor koji se ne nalazi direktno u arhitekturi sistema već je on taj kome će se svaki čvor javljati pre samog ulaska u arhitekturu. On služi kako bi se čvorovi koji žele da uđu u mrežu povezali sa istom. On čuva listu aktivnih čvorova kako bi čvoru koji mu se javio mogao da prosledi nasumičan čvor kome će se javiti.

Arhitektura mreže

Ukoliko mrežu čine manje od 5 čvorova, ona je kompletan graf. Ukoliko je čini više od 5 čvorova, svaki čvor u mreži je povezan sa dva čvora ispred njega i dva čvora ispred njega. U pitanju je dvosmerna komunikacija. Evo i primera grafa sa sedam čvorova.



Konkurentnost

U distribuiranim sistemima se susrećemo sa problemom konkurentnosti, čvorovi ne koriste istu memorije i iste niti pa je samim tim teže izvršavati konkurentne zahteve. U našem sistemu uz implementaciju Suzuki-Kasami algoritma jeste rešen taj problem tako što će samo jedan čvor u trenutku imati token. Čvorovi će samo uz pomoć tokena moći da izvršavaju kritične sekcije koje zahtevaju konkurentnost.

Konstante

Pre pokretanja samog sistema, imamo predefinisane konstante kao što su IP adresa i port bootstrap čvora, radni koren čvorova odakle će moći da ubacuju fajlove u mrežu, donja granica otkaza u milisekundama i gornja granica otkaza

Komande čvorova

1. „info“ komanda – ispis svih mojih trenutnih komšija, da li imam token kod sebe, koliko sam kritičnih sekcija uradio, koliko sam zahteva za kritične sekcije primio i spisak svih fajlova koje čvorovi čuvaju

2. „add name“ komanda – umesto name upisujemo ime fajla ili direktorijuma kojeg želimo da preuzmemo iz radnog korena, ukoliko je u pitanju direktorijum, svi fajlovi iz direktorijuma će biti preuzeti. Prilikom dodavanja fajla potrebna nam je konkurentnost kako bismo ostalim čvorovima rekli da se ovaj fajl nalazi na našem čvoru

3. „pull name“ komanda – umesto name upisujemo ime fajla kojeg želimo da povučemo sa nekog čvora, naš sistem će proći kroz listu fajlova koju čuva svaki čvor i odrediće sa kog čvora trebamo da preuzmemo fajl

4. „remove name“ komanda – uklanja fajl sa našeg čvora i obaveštava ostale čvorove o tome

Otkaz čvorova

Programi su skloni otkazima i samim tim moramo voditi računa o fajlovima koji su u sistemu i ne želimo da oni budu izbrisani iz sistema.

Backup funkcionalnost

Svaki čvor nakon dodavanja fajla će napraviti backup kopiju fajlova. Čvor će poslati poruku svom prethodniku po ID-ju poruku o nazivima fajlova koje je uneo u sistem i zatim će ovaj napraviti „Backup“ folder sa tim fajlovima.

Detektovanja otkaza

Detektovanje otkaza radimo uz pomoć Buddy sistema uz pomoć kojeg je svaki čvor zadužen da proveri stanje čvora ispred sebe i obavesti ceo sistem ukoliko je došlo do pada tog čvora. Ukoliko čvor otkaže, na našem čvoru ćemo odraditi Backup fajlova za taj čvor.

Startovanje Bootstrap čvora

Prilikom startovanja bootstrap čvora, čvor će automatski biti aktivan i osluškivaće nadolazeće poruke, komandom „stop“ se zaustavlja njegov rad.

Startovanje čvora i ulazak u mrežu

Pre samog startovanja programa za ostale čvorove potrebno je da kao argumente postavimo IP adresu i port na kome će slušati ovaj čvor npr. „localhost 1200“, komandom „hello“ čvor će poslati poruku Bootstrapu kako bi se javio da želi da uđe u mrežu. Nakon toga će čvoru biti dodeljen ID u zavisnosti od toga koliko čvora već postoje u mreži i on će biti povezan sa svojim komšijama iz arhitekture.

Komunacija

*primer komunikacije ulaska čvora*

Novi čvor šalje HelloToBootstrap ka boostrapu, bootstrap odgovara porukom HelloFromBootstrap gde mu šalje podatak da li je on prvi čvor u mreži ili ipak treba da se javi nekom drugom čvoru.

Ukoliko treba da se javi drugom čvoru, tad mu šalje poruku HelloToNode tom čvoru, dok kao odgovor dobija HelloFromNode gde dobija podatke o samoj arhitekturi kao što su dosadašnje obrađene kritične sekcije, koji čvor čuva koje fajlove i listu čvorova na osnovu kojeg će zaključiti svoj novi ID. Kada čvor bude ubačen u mrežu on šalje NewNodeToBootstrap poruku ka bootstrapu kako bi ovaj ažurirao svoju listu aktivnih čvorova.

*primer komunikacije dodavanja fajla u mrežu i ulaska u kritičnu sekciju*

Kada čvor primi komandu za dodavanje fajla u mrežu on će početi izvršavanje kritične sekcije kako bi obavestio sve čvorove da se desila promena u sistemu. On to radi uz pomoć Broadcastovanja poruke UpdateSystem u kojoj će naznačiti koje fajlove je ubacio u sistem. Nakon što ostali čvorovi ažuriraju svoj sistem tada će poslati poruku SystemUpdated. Čvor koji je dodao fajl će sačekati da dobije od svih čvorova odgovor da su ažurirali sistem kako bi napustio kritičnu sekciju.

*primer komunikacije povlačenja fajla sa drugog čvora*

Fajl koji zahteva povlačenje fajla pronalazi odakle treba da povuče fajl i šalje poruku PullFile i naziv fajla ka tom čvoru, taj čvor kad primi tu poruku zapakuje taj fajl u listu bajtova kontenta i pošalje ga nazad tom prvom čvoru kako bi ga sačuvao.

*primer komunikacije backup sistema*

Prilikom dodavanja fajlova u sistem, potrebno nam je da uradimo Backup, to radimo slanjem poruke CreateBackup ka čvoru iza nas jer će on primetiti ukoliko naš čvor otkaže. U samu poruku ćemo staviti nazive fajlova koje mu šaljemo kao i njihov sadržaj.

*primer komunikacije buddy sistema*

Ovaj sistem šalje Ping poruku ka čvoru nakon sebe i čega njegov odgovor Pong, ukoliko ne dobije poruku Pong u roku od donjeg limita otkaza, on prelazi u drugu fazu gde će se obratiti čvoru koji se nalazi nakon njegovog čvora i poslati mu poruku IsHeOK koja se odnosi na čvor koji je trebao da odgovori sa Pong. Taj treći čvor će poslati poruku ka drugom čvoru AreYouOK, ukoliko mu ovaj odgovori kroz tri sekunde sa OK porukom, onda će i ovaj vratiti HeIsOK poruku pa provera ide ponovo u krug, ukoliko mu ne odgovori u roku od tri sekunde, ovaj šalje HeIsNotOK poruku i onda će prvi čvor znati da je ovaj čvor otkazao i vrši reorganirazaciju same arhitekture kroz kritičnu sekciju.