Snapshots

Domaći 2

1 Pregled zadatka

Zadatak je u velikoj meri zasnovan na primerima koji su rađeni na vežbi 8. Preporučuje se temeljno izučavanje tih primera pre čitanja teksta zadatka.

Implementirati distribuiran sistem koji podržava sledeće funkcionalnosti:

- Implementiran u Java ili go programskom jeziku. Dozvoljena je upotreba pomoćnih biblioteka za komunikaciju između čvorova.
- Potpuno asinhrona ne-FIFO komunikacija
- Proizvoljan broj čvorova koji su povezani na proizvoljan način (graf nije kompletan). Ako bilo koji čvor bilo kada ima potrebu da pošalje neku poruku svim čvorovima u sistemu, to mora da funkcioniše isključivo preko veza koje su date u konfiguraciji sistema.
- Svaki čvor ima svoj port na kojem prihvata poruke od svojih suseda, i svi slušaju na localhost.
- Radi se jedan snapshot u jednom trenutku, na nivou čitavog sistema. Sistem treba da podrži izradu novog snapshota na proizvoljnom čvoru, nakon što je prethodni snapshot završen.
- Komunikacija sa korisnikom preko CLI ili tekstualnih (skript) datoteka.

Dozvoljeno je koristiti projekat sa vežbi kao polaznu tačku za izradu domaćeg zadatka, kao i implementirati čitav sistem od nule, dok god zadovoljava sve funkcionalne i nefunkcionalne zahteve.

Funkcionalni zahtevi za sistem su opisani u odeljku 2.

Nefunkcionalni zahtevi za sistem su opisani u odeljku 3.

Bodovanje zadatka, kao i instrukcije za predaju zadatka su dati u odeljku 4.

2 Funkcionalni zahtevi

Svaki čvor u sistemu počinje sa unapred određenom količinom bitcake-ova, kao u primerima sa vežbe 8.

U okviru konfiguracije za čvor se uvodi novi atribut “initiators”, koji sadrži listu čvorova koji mogu da budu inicijatori za snapshot u našem sistemu. Samo čvorovi koji su inicijatori će započinjati snapshot.

Neophodno je da bude moguće napraviti snapshot sistema sa proizvoljnog inicijatorskog čvora. Inicijatori moraju da podrže sledeće:

1. Iniciranje više snimaka na istom čvoru, s time da će se obavljati samo jedno snimanje u jednom trenutku sa istog čvora. Višestruki snimci sa istog čvora treba da budu implementirani pomoću Li varijacije Lai-Yang algoritma.
2. Dozvoljeno je da više različitih inicijatora započne snimanje konkurentno. Rezultati ovih snimaka treba da se kombinuju pomoću Spezialetti-Kearns algoritma.

Svi čvorovi, bili inicijatori ili ne, će na osnovu komandi od korisnika da veoma često razmenjuju svoje bitcake zalihe. Rezultat snapshot algoritma treba da bude trenutno bitcake stanje u sistemu, kao što je bio slučaj na vežbi 8 kod Lai-Yang algoritma.

Dozvoljeno je da se svi čvorovi startuju na istoj mašini i slušaju na različitim portovima vezani na localhost.

Dozvoljeno je da čvor A pošalje poruku čvoru B ako je zadovoljeno jedno od sledeća dva:

- Čvorovi A i B su neposredni susedi po konfiguraciji sistema.
- Čvor B je inicijator snapshot algoritma u nekom susednom regionu. Čvor A je indirektno dobio poruku u kojoj se nalazio id čvora B.

Neophodno je da sistem podrži “skriptovano” pokretanje više čvorova, gde se komande za svaki čvor čitaju iz tekstualne datoteke, i izlazi za svaki čvor se pišu u zasebnim datotekama.

3 Nefunkcionalni zahtevi

Neophodno je imati podršku za sledeće situacije, sa navedenim rešenjima. Probleme koji nisu navedeni u tabeli nema potrebe rešavati.

R. Br.	Problem	Rešenje
1.	Korisnik traži izradu snapshota na čvoru koji nije inicijator.	Prijaviti grešku na konzoli i nastaviti normalno sa radom.
2.	Korisnik traži izradu snapshota na čvoru koji je inicijator, ali inicijator nije završio pravljenje prethodnog snapshota.	Prijaviti grešku na konzoli i nastaviti normalno sa radom.

Svi navedeni problemi treba da se reše graciozno, tj. da sistem nastavi normalno da funkcioniše.

Kada inicijator ponavlja snimanje, neophodno je da se slanje rezultata snimanja realizuje diferencijalno, tj. da se na nivou sistema počne “nova” istorija za inicijator X nakon što X prikupi snapshot (Li algoritam).

Neophodno je da se pri sakupljanju snapshota razmeni ukupno O(e) poruka (gde je e broj veza u sistemu) čak i kada ima više konkurentnih inicijatora (Spezialetti-Kearns algoritam).

Pomoću konfiguracione datoteke se navodi:

- Koliko čvorova ima u sistemu.
- Port na kojem svaki čvor sluša.
- Lista suseda za svaki čvor.
- Lista inicijatorskih čvorova u sistemu.

Na svim čvorovima treba da postoje poruke ispisa koje daju informacije o tome koji čvor je roditelj trenutnom u razapetom stablu koje se formira pri sakupljanju rezultata. Ako je čvor inicijator, pri sakupljanju rezultata konkurentnih snapshota treba ispisivati i podatke koji su prikupljeni u svakoj rundi razmene rezultata.

4 Predaja zadatka

4.1 Način predaje zadatka za studente koji prvi put slušaju predmet

Zadatak se predaje slanjem putem Git Classroom-a.

Rok za predaju svih grupa je 12.5.2024. i Github Classroom link je:
https://classroom.github.com/a/KVNs9mYD

Ako nemate nalog na GitHub-u, morate ga napraviti. Ako već imate nalog, prijavite se na GitHub sa Vašim postojećim nalogom.

Nakon što prihvatite zadatak, GitHub će automatski kreirati repozitorijum za Vas sa odgovarajućim imenom i postavkama za zadatak.

Da biste radili na zadatku, morate klonirati repozitorijum na svoj lokalni računar. To možete uraditi koristeći komandu git clone u terminalu sa linkom do Vašeg repozitorijuma.

Nakon što klonirate repozitorijum, možete početi da radite na zadatku lokalno na svom računaru.

4.2 Način predaje zadatka za ponovce

Zadatak se predaje putem mail-a na ml_jovanovic@raf.rs. Java projekat imenovati na sledeći način: “kids_d2_ime_prezime_ind”. Npr. “kids_d2_student_studentic_rn0101”.

Arhivirati ovaj direktorijum (.zip), okačiti na svoj drive i u mail-u poslati link ka arhivi, pošto će Google mail verovatno blokirati direktan attachment.

U tekstu mail-a obavezno navesti:

- Ime i prezime
- Broj indeksa
- Grupa, po zvaničnom spisku
- Programski jezik (java / go) i razvojno okruženje (Eclipse / Intellij)

Subject mail-a mora da bude u obliku: “[KiDS 2024] D2 ime_prezime_ind”.

Npr. “[KiDS 2024] D2 student_studentic_rn0101”

Naziv arhive mora da bude u obliku: “kids_d2_ime_prezime_ind.zip”

Npr. “kids_d2_student_studentic_rn0101.zip”

4.2 Odbrana i bodovanje

Odbrana domaćih zadataka je obavezna. Termin za odbranu drugog domaćeg zadatka će biti naknadno objavljen. Ako ste iz bilo kog razloga sprečeni da prisustvujete odbrani, obavezno to najavite što pre, kako bismo mogli da zakažemo vanredni termin za odbranu.

Svrha odbrane je da se pokaže autentičnost zadatka. Ovo podrazumeva odgovaranje na pitanja u vezi načina izrade zadatka, ili izvršavanje neke izmene nad zadatkom na licu mesta. U slučaju da odbrana nije uspešna, dodeljuje se -15 poena na domaćem zadatku.

VAŽNO: Studiranje i istraživanje algoritama kroz korišćenje koda pronađenog na internetu je dozvoljeno i može vam pomoći u boljem razumevanju materije. Međutim, ukoliko više studenata koristi identičan izvorni kod pronađen na internetu, to će biti smatrano kao korišćenje istog koda i svim uključenim studentima će biti dodeljeni negativni poeni. Ista pravila važe i za korišćenje algoritama generisanih veštačkom inteligencijom.
Cilj domaćih zadataka na akademskim studijama je samostalno istraživanje i sticanje znanja bez oslanjanja na rešenja sa interneta ili rešenja generisanih od strane veštačke inteligencije.

Zadatak se boduje na sledeći način:

- Moguće je praviti više snimaka sa istog čvora - 4 poena
- Implementacija stabala i regiona pri konkurentnom snimanju - 6 poena
- Sumiranje rezultata iz različitih regiona - 5 poena

Zadatak je moguće raditi parcijalno, ali obavezno je da se kompajluje i da može da se pokrene, kao i da je moguće pokazati da implementirana stavka ispunjava funkcionalne i nefunkcionalne zahteve.