

Stefan Anić

Email: sanic16@raf.rs

Broj indeksa: RN 19/16

Grupa: 401

Game of Chaos

4th June 2020.

UVOD

Ovim dokumentom se opisuje distribuirani sistem koji računa fraktalne strukture primenom igre haosa. Opisan je način funkcionisanja sistema, protokol razmene poruka, sastav poruka, način upotrebe i algoritam izračunavanja.

Distribucija se ogleda u podeli posla. U ovom sistemu, posao je predstavljen izračunavanjem tačaka u nekom dodeljenom regionu. Skup tačaka formira fraktalnu sliku. Distribucija posla se oslanja na samoorganizaciju čvorova u sistemu.

Sistem se zasniva na mrežnoj komunikaciji između više procesa u svrhu podele zadatog posla.

IDENTIFIKACIJA ČVOROVA

U sistemu učestvuju dve vrste čvorova:

1. Bootstrap
2. Servent (u daljem tekstu, samo čvor).

Bootstrap

Bootstrap predstavlja komponentu pomoću koje ostali čvorovi bivaju registrovani u sistem. Podrazumeva se da je Bootstrap u konstantnom radu, kao i to da su svi ostali čvorovi svesni njegovog odredišta (IP adrese i porta).

Njegova uloga je evidentiranje svih aktivnih čvorova u sistemu. Shodno tome, svi ostali čvorovi iniciraju svoju konekciju najpre sa njim, kako bi dobili odredište (IP i port) ka serventu koji se nalazi na začenu liste registrovanih čvorova, a koji će ih dalje uputiti kako da izvrše registraciju i dobiju svoj identifikator. Nakon dobijenog identifikatora, čvor je dužan da obavesti Bootstrap o uspešnom povezivanju u sistem.

Kako bi se ta lista registrovanih čvorova održavala, neophodno je da svaki od čvorova prilikom svog izlaska javi bootstrap-u da je napustio sistem.

Servent

Servent je opisan jedinstvenim identifikatorom koji mu sistem dodeljuje, IP adresom i portom. Identifikator zavisi od njegove pozicije u sistemu i rasporedu drugih čvorova.

On predstavlja čvor u sistemu koji ima mogućnost obavljanja posla - izračunavanja fraktala. Celokupan sistem serventa je u stanju da obavlja samo jedan posao u trenutku.

Pored toga, može delovati i kao klijent koji ima mogućnosti da započne posao, stopira posao i da na zahtev prikuplja izračunate rezultate drugih servenata u sistemu na osnovu kojih generiše PNG sliku fraktala.

ORGANIZACIJA ČVOROVA I NAČIN RAZMENE PORUKA

Algoritam za povezivanje

Prilikom povezivanja novog čvora u sistem, dodeljuje mu se ID za 1 veći od prethodno dodatog čvora. Novi čvor postaje sledbenik poslednjeg dodatog čvora, a sledbenik novog čvora će postati prvi čvor dodat u sistem (sa ID-em 0).

Novododati čvor će imati N suseda. N je broj za koji važi da je prvi 2^N manji od broja trenutno aktivnih čvorova. Te susede ćemo izračunati na sledeći način: $i=[0..N]$, svaki čvor za čiji ID važi sledeće: $id_{NovogČvora} + 2^i \bmod brojTrenutnoAktivnihČvorova$, postaće sused novog čvora.

Algoritam za rutiranje

Prethodno opisanim algoritmom se formira struktura koja će se koristiti za dalje rutiranje poruka, a kojom ćemo izbeći da čvorovi direktno komuniciraju (formiraju kompletan graf).

Poruke se šalju korišćenjem IP adrese i porta drugog čvora, ali putanja do cilja nije nužno direktna. Iako servent može imati utisak da je putanja direktna, prilikom slanja poruke proverava se da li je čvor sa ciljanom IP adresom i portom sused pošiljaoca, ukoliko jeste, poruka se šalje direktno. U suprotnom, ukoliko čvor nije sused pošiljaocu, poruka će se poslati susednom čvoru sa prvim manjim ID-em od primaoca.

Čvor koji primi poruku koja nije za njega, tj. koja je upućena ka drugoj IP adresi i portu, po istom principu, poruku prosleđuje dalje.

RASPODELA POSLA

Prilikom pokretanja posla ili prilikom izmene broja aktivnih čvorova u sistemu, obavlja se raspodela (preraspodela) posla između čvorova.

Čvor na kome je pokrenuta raspodela posla je zadužen za izračunavanje koliko drugih čvorova će učestvovati u obradi posla kao i za generisanje fraktalnih ID-eva koji će biti dodeljeni učesnicima.

Nakon toga, za fraktalne ID-eve na svom nivou, određuje krajnje tačke poligona: iterira kroz sve tačke inicijalnog poligona međusobno i računa tačke poligona za naredni nivo po formuli: $newCoordinate = point1.coordinate + proportion * (point2.coordinate - point1.coordinate)$. Tako izračunate tačke, kao i podfraktalne ID-eve, šalje jednom od čvorova koji će rekurzivno ponavljati isti posao dok svaki od čvorova ne dobije jedan fraktalni ID kada će započeti sa računanjem tačaka u okviru svojih granica.

Ukoliko smo izvršili preraspodelu (nakon dodavanja ili gašenja čvora), neke tačke već postoje u sistemu i treba ih proslediti prilikom startovanja. U ovom slučaju imamo dve moguće situacije:

1. Situacija kada je proporcija ≤ 0.5 : U čijem slučaju uzimamo tačke koje su unutar poligona novokreiranih regiona (označenim fraktal ID-em) i prosleđujemo ih u poruci za započinjanje posla.
2. Situacija kada je proporcija > 0.5 : U ovom slučaju ne možemo porediti da li je tačka unutar poligona, već ćemo tačke dodeliti čvorovima nasumično.

ALGORITAM IZVRŠAVANJA POSLA

Nakon započetog posla, čvor kreira novu niti i započinje izračunavanje tačaka fraktala po sledećem principu: bira se nasumična tačka regiona - P2, a nakon toga jedna od K inicijalnih (graničnih) tačaka regiona - P1. Na osnovu te dve tačke, računamo novu tačku P(x, y) po formuli $P.coordinate = P1.coordinate + proportion * (P2.coordinate - P1.coordinate)$

Poslednju izračunatu tačku pamtimo kao P2 i u narednoj iteraciji postupak ponavljamo počev od odabira neke od K graničnih tačaka - P1.

Novokreirane tačke smeštamo u set kako bi čuvali samo jedinstvene tačke i smanjili popunjenost memorije.

KOMUNIKACIJA

Komunikacija servera ↔ bootstrap

Komunikacija sa Bootstrap-om se odvija pomoću socket-a pomoću kog se obrađuju poruke sledećih formata:

1. Hail - inicijalna poruka

Format zahteva: `Hail\nip_adresa_novog_čvora:port_novog_čvora\n`

Format odgovora: `ip_adresa_čvora_u_sistemu:port_čvora_u_sistemu\n`

Izuzetak je prvi čvor u sistemu koji kao odgovor dobija: `:-1\n` na osnovu čega će znati da sebi može dodeliti id=0 i ne treba se dodatno javljati bootstrapu radi evidencije.

2. New - poruka koju svi čvorovi (osim prvog) šalju Bootstrap-u nakon uspešnog povezivanja u sistem, radi dalje evidencije

Format zahteva: `New\nip_adresa_čvora:port_čvora\n`

3. Quit - poruka koja obaveštava bootstrap da je čvor napustio sistem

Format zahteva: `Quit\nip_adresa_čvora:port_čvora\n`

Komunikacija servera ↔ server (poruke)

Komunikacija u sistemu se obavlja putem socket-a. Svaki čvor je označen svojom IP adresom i portom na kom osluškuje nove poruke. Poruke su serijalizovane upotrebom Javine serijalizacije objekata i svaka od poruka je opisana svojim tipom, IP adresom i portom pošiljaoca, opcionim tekstom, svojim jedinstvenim identifikatorom, IP adresom i portom primaoca, kao i putanjom svog kretanja kroz sistem kojom je došla do odredišta. Pored svog osnovnog sadržaja, a zavisno od tipa poruke, poruka može sadržati i dodatne parametre. Prilikom razmena poruka ili iniciranja komandi, treba povesti računa da ne dođe do konkurentnih pristupa koji u ovoj implementaciji nisu podržani.

Objekti koji se mogu naći u poruci, pored primitivnih tipova, mogu biti i instance klase:

1. ServentInfo - klasa koja sadrži *id* čvora, *ipAddress*, *listenerPort*, kao i listu konfiguracija mogućih poslova - *jobs*
2. Job - klasa koja opisuje neki posao i sadrži *name*, *proportion*, *width*, *height*, *initialPointsCount* i *points* - listu inicijalnih tačaka

Poruke koje se razmenjuju između servera su definisane sledećim tipovima:

NEW_NODE

Opis: Poruka koja javlja našem prethodniku (poslednjem čvoru do tog trenutka) da se želimo uključiti u sistem.

Handler: Pošiljaocu poruke dodeljuje ID za 1 veći od svog i obaveštava ga o tome putem poruke tipa WELCOME u koju smešta i odredište (IP adresu i port) prvog čvora u sistemu.

WELCOME

Opis: Poruka koja obaveštava novi čvor da se uspešno povezao u sistem.

Dodatni sadržaj:

- firstServentIpAndPort - odredište ka prvom čvoru u sistemu, primer: `192.160.0.1:1000`

Handler: Inicijalizuje listu suseda novog čvora, obaveštava Bootstrap o uspešnoj registraciji porukom New. Nakon toga, prvom čvoru u sistemu šalje poruku tipa UPDATE, čime se inicira ažuriranje suseda ostalih čvorova, a kasnije (kada poruka obiđe krug) i svojih.

UPDATE

Opis: Poruka koja obaveštava sve aktivne čvorove o izmenama u sistemu prilikom uključivanja novog čvora. Poruka se inicira u handler-u WELCOME poruke i prosleđuje se sledbeniku. Svaki čvor je zadužen da po prijemu ove poruke ažurira svoje susede i evidentira novododati čvor, te istu prosledi dalje, uz sve svoje evidentirane čvorove kao dodatni sadržaj.

Pored toga, ukoliko se neki posao izvršavao, u nju se smeštaju već izračunate tačke kako bi se kasnije posao mogao nastaviti nakon preraspodele.

Dodatni sadržaj:

- nodesInfo - mapa evidentiranih čvorova od strane pošiljaoca. Ključ mape je ID čvora, a vrednost ServentInfo objekat čvora
- activeJob - ukoliko se neki posao obavljao pri kreiranju ove poruke, instanca tog posla se prosleđuje
- computedPoints - ukoliko se neki posao obavljao u sistemu, poruka će sadržati set već izračunatih tačaka svakog čvora

Handler: Na osnovu nodesInfo mape, ažurira se svest o aktivnim čvorovima u sistemu. Ukoliko je čvor obavljao neki posao u trenutku obrade UPDATE poruke, on je dužan da svoje izračunate tačke doda na poruku, zaustavi izradu posla i poruku prosledi dalje, sa informacijama o čvorovima kojih je on svestan u tom trenutku. Kada poruka obiđe krug (vrati se na svog inicijatora), ukoliko se posao obavljao u sistemu, inicijator je dužan da započne posao ponovo i rasporedi već izračunate tačke radnicima.

QUIT

Opis: Pre nego da čvor napusti sistem, neophodno je da o tome budu obavešteni svi učesnici, kao i Bootstrap server. Kako sistem nije otporan na otkaze, ova akcija je obavezna radi pravilnog funkcionisanja. Pri izlasku iz sistema, čvor je dužan da svom sledbeniku posalje QUIT poruku i time inicira svoje gašenje. Takođe, ukoliko je čvor obavljao neki posao, neophodno je da ga zaustavi i svoje izračunate tačke prosledi u tu QUIT poruku.

Dodatni sadržaj:

- `quiterId` - ID čvora koji je inicirao svoje gašenje
- `activeJob` - Ukoliko se neki posao obavljao pri kreiranju ove poruke, instanca tog posla se prosleđuje
- `computedPoints` - Ukoliko se neki posao obavljao u sistemu, poruka će sadržati set već izračunatih tačaka svakog čvora

Handler: Prilikom primljene poruke, primalac je dužan da ukloni čvor sa prosleđenim `quiterId`-em iz svoje mape poznatih čvorova i dekrementira ID-eve svih poznatih čvorova većih od `quiterId`-a, uključujući i svoj ID. Nakon toga, zbog promene ID-eva čvorova u sistemu, neophodno je izvršiti ažuriranje suseda. Ukoliko je čvor obavljao neki posao u trenutku obrade QUIT poruke, on je dužan da svoje izračunate tačke doda na poruku, zaustavi izradu posla i poruku prosledi dalje. Kada poruka obiđe krug, ukoliko se posao obavljao u sistemu, neophodno je ponovo pokrenuti posao i rasporediti već izračunate tačke radnicima.

START_JOB

Opis: Svaki od aktivnih čvorova može započeti izračunavanje nekog posla u sistemu (ukoliko u sistemu ne postoji već započet posao). Inicijator je dužan da raspodeli posao i obavesti čvorove da mogu početi sa radom. Takođe, posao se može inicirati i sa već izračunatim tačkama (preraspodela nekon što se neki čvor uključi ili isključi). Raspodela posla je opisana [ovde](#). Nakon pokretanja posla, prvom čvoru u sistemu se šalje START_JOB poruka.

Dodatni sadržaj:

- `job` - instanca posla čije izračunavanje treba započeti
- `level` - fraktalni nivo slike
- `fractalIds` - fraktalni ID-evi izračunati pri inicijalizaciji posla koje primalac koristi da podeli posao čvorovima na svom nivou
- `initialPoints` - granične tačke (pod)poligona
- `fractalIdToNodeIdMap` - mapa koja predstavlja koji čvro će biti zadužen za koji fraktalniID, izračunata pri inicijalizaciji posla
- `precomputedPoints` - Ukoliko se neki posao obavljao u sistemu, poruka će sadržati set već izračunatih tačaka svakog čvora

Handler: Čvor koji primi ovu poruku je dužan da uradi raspodelu posla sa čvorovima na svom fraktalnom nivou, pošalje im START_JOB poruku i započne svoje izračunavanje. Ukoliko je poruka sadržala `precomputedPoints`, te tačke će se posmatrati kao već izračunate.

STOP_JOB

Opis: Svaki od aktivnih čvorova može inicirati stopiranje započetog posla. Ime započetog posla se prosleđuje kroz STOP_JOB poruku prvom čvoru u sistemu.

Dodatni sadržaj:

- jobName - naziv posla kog želimo završiti

Handler: Primalac poruke je zadužen da stopira izvršavanje prosleđenog posla i ukloni izračunate tačke. Takođe, poruku treba proslediti svom sledbeniku dok poruka ne dođe do poslednjeg čvora koji je radio izračunavanje, u tom slučaju zaustavljamo prosleđivanje. Ukoliko je ime posla nepoznato čvoru, inicijatoru stopiranja se šalje poruka tipa ERROR uz obaveštenje.

ASK_FOR_RESULT

Opis: Svaki od aktivnih čvorova može inicirati generisanje rezultata posla. Generisanje rezultata je moguće izvesti za celokupan posao ili za određeni deo posla, uz prosleđen fraktal ID.

Dodatni sadržaj:

- jobName - ime posla čiji rezultat zahtevamo
- fractalId - opcioni parametar konkretnog fraktalnog ID-a čiji rezultat tražimo. Ukoliko nije prosleđen, rezultat se računa na nivou celog sistema
- appendedResults - set izračunatih tačaka u koji svi radnici ubacuju svoje rezultujuće tačke

Handler: Primalac poruke je dužan da svoje rezultate smesti u appendedResults i poruku prosledi svom sledbeniku koji će uraditi isto. Poruku prosleđujemo dalje dok ona ne dođe do poslednjeg čvora koji radi izračunavanje i koji će prikupljene rezultate proslediti inicijatoru. Ukoliko je u poruci naznačen fractalId, čvor će svoje rezultate odmah proslediti inicijatoru. Ukoliko je ime posla nepoznato čvoru, inicijatoru rezultata se šalje poruka tipa ERROR uz obaveštenje.

RESULT

Opis: Nakon prikupljanja izračunatih rezultata, oni se šalju inicijatoru rezultata kroz RESULT poruku.

Dodatni sadržaj:

- jobDetails - posao čiji su se rezultati prikupljali
- resultPoints - set prikupljenih rezultata
- fractalId - opcioni parametar ukoliko su se rezultati zahtevali za specifičan fraktal ID

Handler: Nakon prikupljenih rezultata, inicijator je dužan da iste iscrta i sačuva u PNG fajl.

ASK_FOR_STATUS

Opis: Svaki od aktivnih čvorova može inicirati uvid u trenutno stanje posla. Trenutno stanje posla je moguće zatražiti za celokupan posao ili za određeni deo posla, uz prosleđen fraktal ID.

Dodatni sadržaj:

- jobName - ime posla
- fractalIdToPointCountMap - mapa koja čuva koliko je tačaka izračunao čvor sa specifičnim fraktalnim ID-em
- fractalId - opcioni parametar ukoliko se stanje zahteva za specifičan fraktal ID

Handler: Primalac poruke je dužan da svoje stanje smesti u fractalIdToPointCountMap i poruku prosledi svom sledbeniku koji će uraditi isto. Poruku prosleđujemo dalje dok ona ne dođe do poslednjeg čvora koji radi izračunavanje i koji će prikupljena stanja proslediti inicijatoru. Ukoliko je u poruci naznačen fractalId, čvor će svoje stanje odmah proslediti inicijatoru. Ukoliko je ime posla nepoznato čvoru, inicijatoru statusa se šalje poruka tipa ERROR uz obaveštenje.

STATUS

Opis: Nakon prikupljanja statusa, stanja aktivnih čvorova se šalju inicijatoru stanja kroz STATUS poruku.

Dodatni sadržaj:

- jobName - ime posla
- fractalIdToPointCountMap - mapa koja čuva koliko je tačaka izračunao čvor sa specifičnim fraktalnim ID-em

Handler: Nakon prikupljanja statusa svih čvorova koji obrađuju posao, primalac je dužan da ista prikaže na konzoli.

ERROR

Opis: Ukoliko se u sistemu desi nešto neočekivano, drugi čvor o tome obaveštavamo korišćenjem ERROR poruke.

Dodatni sadržaj:

- text - tekstualni opis greške

Handler: Nakon dobijene ove poruke, primalac je dužan da kroz error stream korisniku prikaže dobijeni tekst.