

Fraktal - Dokumentacija

Uvod

Ovaj dokument predstavlja tehničku dokumentaciju za distribuirani sistem koji služi za distribirano računanje i čuvanje fraktala po pravilu igre haosa. Sistem podržava i esportovanje slike u direktorijum kao jpg, gde se sve operacije mogu videti kao vizuelni prikaz celog ili dela posla. Sve datoteke i direktorijumi su jednoznačno određeni svojom relativnom putanjom.

Podržane operacije su:

- **start X** - započinje posao sa nazivom x
- **status X [id]** - dohvata status za ceo posao ako fraktalni id čvora nije naveden, ako jeste dohvata se broj izračunatih fraktala za taj posao.
- **stop** - izlazak iz mreže.
- **result X[id]** - dohvata sve fraktale za neki posao ako id nije naveden i eksportuje ih u png, ako id jeste naveden onda će dohvatiti fraktale tačno za traženog serventa i eksportovaće njegove izračunate fraktale u png.
- **quit X** - završava tačno navdeni posao..

Konfiguracija

Svaki čvor u sistemu ima svoju konfiguracionu datoteku (.properties fajl) preko koje se podešavaju osnovni parametri tog čvora. Prilikom pokretanja čvora, kao argument se mora proslediti putanja do konfiguracione datoteke. Ona mora da sadrži:

- Port na kojem će čvor da sluša. (short)
- IP adresa i port bootstrap čvora - odeljak (string i short)
- Slaba granica otkaza - odeljak (int)
- Jaka granica otkaza - odeljak. (int)

● Skup predefinisanih poslova. Svaki posao ima sledeće attribute:

- Naziv - jedinstveno simboličko ime za ovaj posao. (string)
- N - broj tačaka fraktalne strukture. (int, $3 \leq N \leq 10$)
- P - udaljenost između trenutne tačke i odredišta na kojoj će se pojaviti nova tačka.

(double)

- W, H - dimenzija površine na kojoj se računaju tačke. (dva int-a)
- A - skup N tačaka. (niz od N parova int-ova)

Bootstrap

Bootstrap server mora biti konstantno prisutan (pokrenut) dok traje rada sistema. On je zadužen za dostavljanje prvog kontakta čvoru koji se prijavi kao da želi da se priključi nekom poslu (start). Čvor šalje bootstrap serveru **hail** poruku, nakon koje bootstrap server ili odgovara porukom **newNode** i šalje odgovarajućem čvoru u mreži sve čvorove koji treba da se povežu ili ih ubacuje u idle listu ako ne mogu da se stave na izvršavanje. Bootstrap u sebi ima lokalnu listu aktivnih čvorova, idle čvorova i deleted čvorova, i na osnovu mapping mape zna kada je moguće da idle čvorovi postanu aktivni i prosledjuje ih odgovarajućem čvoru koji ih kasnije uvezuje i aktivira. Za bootstrap poruke, neophodne su tcp konekcije. Onu sebi sadrži liste za svaki posao koji je aktivan.

Poruke

Base message

Svaka poruka je u suštini **message**, i ona obavezno ima:

- Sender - čvor koji šalje poruku - *SenderInfo
- Receiver - čvor koji prima poruku - *SenderInfo
- ID - id poruke u mreži - int64
- Type - tip poruke - enum

- pointNum - int koji je zadužen da obelji na koji posao se odnosi
- messageText - string, tekst poruke

Specijalni tipovi poruka

Svi specijalni tipovi poruka su samo proširenja osnove message poruke u zavisnosti od tipa, koji imaju dodatna polja u zavisnosti od potrebe.

Hail

Identična je kao BasicMessage šalje se pri startu nekog serventa služi za javljanje bootstrap-u radi pravilnog rasporeda unutar naše mreže

NewNode

Ovu poruku Bootstrap šalje serventu na koji će se novi serventi uključiti, ona ima listu servenata koji naš "parent" servent treba da zakači na sebe startuje i dodeli njihove nove fraktal id-jeve

Welcome

Ona ima dodatna polja: FraktalNid koji je prosleđen od strane našeg "parent" čvora, listu servenata koji će postati naše komšije, i fraktalni id koji treba da setujemo

Ready

Šalje se našim susedima da smo se setovali kako treba i krenuli sa radom.

Num

Poruka koja vraća trenutni broj aktivnih servenata, njen handler setuje globalni atomic int na trenutni broj aktivnih servenata.

Idle

Ista kao BasicMessage služi nam kada brišemo serventa na komadi quit u slučaju da neki servent treba da se setuje na idle.

Neighbor Quit

Ima u sebi fraktalni id našeg roditelja i listu suseda od čvora koji brišemo da bi javili svim njegovim čvorovima i da bi resetovali fraktalni id roditelja

QuitJob

Ima u sebi serventInfo serventa koji brišemo radi rebalansiranja našeg grafa.

ResultAsk i StatusAsk

U sebi imaju listu servenata kojima smo već poslali poruke radi izbegavanja duplikata.

ResultReply i StatusReply

U sebi imaju listu tačaka za crtanje i fraktalni id

ResultSecific i StatusSpecific

U sebi ima fraktalni id čvora koji tražimo i listu kome smo već poslali.

Detaljan opis rada

Result i Status

Na samom početku ove funkcije šaljemo numMessage koji nam vraća broj trenutnih aktivnih servenata na poslu koji trenutno radimo, čekamo ga unutar while-a koji periodično proverava da li smo dobili broj da bi mogli da nastavimo dalje. Kada dobijemo broj tada možemo svima poslati ask poruku koja pita sve čvorove koa brodkast da nam daju njihove izračunate tačke u slučaju da smo pozvali result ili koliko tačaka je izračunato ako smo pozvali status. Nakon toga čekamo da nam svi čvorovi odgovore unutar while-a koji periodično proverava da li smo dobili odgovor. Kad dobijemo odgovor izcrtavamo našu fraktalnu sliku pomoću dobijenog niza. Ako smo tražili konkretan fraktal sa nekim id-jem onda ćemo preko specific poruka dobiti rezultat.

Quit

Quit sklanja sa serventa koji je pozove sa nekog posle koji je definisan preko prosledjenog imena. Radimo po principu zamene servenata. Nalazimo serventa koje je idle ili ako njega nema koji je zadnji dodat u mrežu, menjamo im njihov fraktalni i nodeid i listu suseda. Nakon

toga je neophodno da udjemo u susede čvora koji smo zamenili da bi resetovali fraktalni id njihovog roditelja i da bi njegove susede proverili da li neki od njih treba da se stavi na idle prilikom rekonstrukcije naše mreže.

Stop

Gasi pokrenute listenere, drawere i elegantno ubija čvor.

Start

Stavlja serenta da radi neki posao salje ga bootstrapu da ga rasporedi.