

Filip Papic RN-127/2018

# Distribuirani haos dokumentacija

## Contents

<b>Uvod .....</b>	<b>2</b>
<b>Arhitektura Sistema .....</b>	<b>3</b>
<b>Cvorovi.....</b>	<b>4</b>
Bootstrap .....	4
Servent.....	5
<b>Konfiguracija čvora .....</b>	<b>6</b>
<b>Komande i izvestaji .....</b>	<b>7</b>
<b>Poruke .....</b>	<b>8</b>
Tipovi poruka.....	8

## Uvod

Ovaj document predstavlja tehnicku dokumentaciju za funkcionalan distribuirani sistem koji računa fraktalne strukture primenom igre haosa. Primer implementirane igre može da se vidi na [ovom](#) linku. Sistem korisniku omogućava sledeće:

- Pokretanje izračunavanja za jedan ili više fraktala.
- Izlistavanje aktivnih čvorova i njihovih aktivnosti.
- Dobijanje kompletne slike ili dela nekog fraktala.
- Otpornost na otkaze.
- Sistem ne mora da podržava istovremeno izvršavanje operacija na različitim čvorovima.
- Takođe ne treba podržavati istovremeno uključivanje ili isključivanje čvorova.

## Arhitektura Sistema

Čvorovi u sistemu treba da konstruišu jednu ili više fraktalnih slika pomoću igre haosa. Za svaki čvor treba da bude jasno definisano koji deo slike on računa. Svakom čvoru treba definisati fraktalni ID kojim se naznačava koji deo izračunavanja on obavlja.

Glavna ideja sistema je da funkcioniše po principu Map-Reduce za deljenje posla(fraktala) na manje delove i njihovo izvršavanje na pojedinačnim cvorovima. Map deo predstavlja pronalazenje posla i njegovo deljenje na manje delove i dodeljivanje tih delova u mapu sa ključevima koji odgovaraju ID-jevima cvorova, Calculate deo predstavlja pojedinačan cvor koji izvršava svoj deo posla i po završetku javlja svima svoj rezultat, dok Reduce deo sklapa parcijalne rezultate i formira konačan rezultat u zavisnosti od želje korisnika. Map i Reduce deo se izvršavaju na **inicijatorskom** cvoru, tj. deljenje posla na manje delove kao i sklapanje delova posla u veci se obavlja na cvoru koji je zatražio Start ili Result komandu.

U Map delu pri podeli posla, gleda se broj trenutno slobodnih cvorova i na osnovu njih se fractal deli na poseban broj poslova. Ako se računa fraktal igrom sa N tačaka, fraktalni ID čvora će biti broj koji se sastoji od cifara od 0 do N-1. Što je dublje čvor ugnežđen u fraktal, toliko više cifara će imati.

Ako je broj čvorova u sistemu takav da posao ne može da se podeli na delove gde svaki čvor ima svoj fraktalni region, onda neki čvorovi mogu da budu idle. Dakle svaki čvor treba da računa jedan ceo fraktal na nekom nivou slike.

## Cvorovi

### Bootstrap

U sistemu postoje cvorovi tj. serventi (server/client) koji mogu da se ukljuuju i iskljuuju u/iz mreze i koji medjusobno komuniciraju. Takodje postoji i poseban, jedan cvor pod nazivom Bootstrap. Bootstrap server mora biti konstantno prisutan (pokrenut) dok traje rada sistema. Bootstrap cvor ima sledece funkcionalnosti:

- Koristi se isključivo za prvo uključivanje i isključivanje čvora u mrežu. Čim bootstrap prosledi novom čvoru adresu i port nekog čvora iz sistema, komunikacija sa bootstrap-om se prekida.
- Bootstrap server ima veoma ograničen protok. Komunikacija sa bootstrap serverom mora biti svedena na minimum.
- Nije dozvoljeno da bootstrap server bude svestan arhitekture sistema, te da on bude taj koji će je organizovati. Sistem mora da bude samoorganizujući.

Bootstrap ima svoj poseban dobro poznat port i IP adresu preko kojih ostali cvorovi komuniciraju sa njim. Ta komunikacija može biti jedna od sledećih:

- Hail – servent šalje ovu poruku Bootstrap-u kada želi da se uključi u sistem. U ovoj poruci se nalazi broj porta i IP adresa serventa. Bootstrap šalje kontakt nekog nasumičnog čvora (port i IP) za koji zna da se nalazi u sistemu, kao i veličinu trenutne liste aktivnih cvorova, koja predstavlja serventov jedinstveni ID. Ako je on prvi servent onda mu Bootstrap odgovara sa -1 -1 0, što znači da je prvi servent u sistemu i da će biti kontakt sledećem serventu koji se bude uključivao.
- New – servent šalje ovu poruku Bootstrap-u da bi obavestio o svom uspesnom uključivanju u sistem i da Bootstrap treba da ga doda u listu aktivnih servenata. U poruci se takodje nalaze port i IP adresa serventa.
- Bye – servent šalje ovu poruku Bootstrap-u kada korisnik unese Quit komandu i servent traži od Bootstrap-a da ga ukloni iz liste aktivnih servenata.

## Servent

Servent ima svoj port, IP adresu i ID u sistemu. Serventi su čvorovi koji komuniciraju sa bootstrapom i međusobno. Dva serventa komuniciraju preko socketa sa svojim portovima i IP adresama, a komunikaciju čine razmena poruka. Svaka poruka ima:

- MessageType - tip poruke (npr. START, RESULT, NEW\_NODE...)
- senderPort - čvor koji šalje poruku
- receiverPort - čvor koji prima poruku
- senderIP – IP adresa koja šalje poruku
- receiverIP – IP adresa koja prima poruku
- messageID - id poruke u mreži
- messageText – opcioni tekst poruke

Poruke mogu da sadrže i dodatne parametre što će se videti u narednom delu dokumentacije.

## Konfiguracija čvora

Pri pokretanju čvora, automatski se isčitava konfiguraciona datoteka u kojoj se navode sledeći atributi:

- Port na kojem će čvor da sluša. (short)
- IP adresa i port bootstrap čvora - (string i short)
- Slaba granica otkaza - (int)
- Jaka granica otkaza - (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 u opsegu 0-1)
  - W, H - dimenzija površine na kojoj se računaju tačke. (dva int-a)
  - A - skup N tačaka. (niz od N parova int-ova)

## Komande i izvestaji

Korisnik može da zada sledeće komande sistemu:

- `status [X [id]]` - Prikazuje stanje svih započetih izračunavanja - broj tačaka na svakom fraktalu. Naznačava za svaki fraktal koliko čvorova radi na njemu, fraktalni ID, i koliko tačaka je svaki čvor nacrtao. Ako se navede X kao naziv izračunavanja, onda se dohvata status samo za njega. Ako se navede posao i fraktalni ID, onda se dohvata status samo od čvora sa tim ID.
- `start [X]` - Započinje izračunavanje za zadati posao X. X može da bude simboličko ime nekog posla navedenog u konfiguracionoj datoteci. Ako se X izostavi, pitati korisnika da unese parametre za posao na konzoli. Proveriti da je ime posla jedinstveno, kao i da su svi parametri validnih tipova. Ako je ovo K-ti posao u sistemu, neophodno je da ima makar K čvorova aktivno. Ako nema K čvorova aktivno, ne startovati posao.
- `result X [id]` - Prikazuje rezultate za završeno izračunavanje za posao X. Korisnik može, a ne mora da navede fraktalni ID za rezultat. Ako se izostavi, onda se dohvata rezultat za ceo posao, u suprotnom samo za taj fraktalni ID. Slika treba da se eksportuje kao PNG.
- `stop X` - Zaustavlja izračunavanje za posao X. Fraktal u potpunosti nestaje iz sistema, i čvorovi se preraspoređuju na druge poslove.
- `quit` - Uredno gašenje čvora.

## Posao

Posao je osnovna jedinica izvršavanja u sistemu. Sadrži svoje ime, broj početnih tacaka i njihove koordinate, visinu i širinu slike na kojoj se crta fraktal kao i proporciju za fraktal. Klase vezane za posao i njihov krati opis:

- JobChaos – služi da izvršava igru haosa na osnovu dodeljenog posla i njegovih koordinata.
- JobResult- služi da na osnovu argumenta iz komande izvuče odgovarajuće rezultate iz mape rezultata po imenu, sabere njihove rezultate i prosledi JobImage-u koji od njih pravi PNG fajl.
- JobImage – služi da iscrta dobijene coordinate iz posla u PNG fajl i doda taj fajl u images folder.
- JobSpread-sluzi da, ukoliko postoji više pokrenutih poslova u isto vreme, podeli broj cvorova ravnopravno i pozove JobSplit za svaki posao.
- JobSplit-sluzi da podeli posao na manje delove u zavisnosti od broja dodeljenih cvorova. Podela se deli po koordinatama, gde nove tacke zavise od proporcije.
- NewJobCreator- služi za unos novog posla preko komandne linije



## Poruke

Svaka poruka ima svoj handler koji izvršava određenu funkciju pri primanju svog tipa poruke. Pod tim funkcijama spada menjanje lokalnih podataka, slanje poruka dalje, ispis...

### Tipovi poruka

**NewNodeMessage** – Kada novi cvor dobije od bootstrapa podatke za cvor postojeći cvor, on tom cvoru šalje ovu poruku. Postojeći cvor po primanju ove poruke novom cvoru šalje Welcome poruku. Ukoliko ima kolizije šalje Sorry poruku.

**WelcomeMessage** – Posto primi NewNode poruku od novog cvora, ovaj cvor novom cvoru šalje Welcome poruku. U handleru poziva init metodu iz ChordState-a kojom postavlja inicijalnu mapu i prvog naslednika tako da možemo da posaljemo Update poruku. Takođe javlja Bootstrap-u da nema sudara (collisions).

-newNodePort (int) - novi port

-newNodeIP (String) – novi IP

-hisValues (Map<Integer, Integer>) – njegove vrednosti za value mapu

**SorryMessage** – Ovaj tip poruke vrši ispis gde javlja da je došlo do sudara i da treba da se promeni port za novi cvor.

**UpdateMessage** – Novi cvor šalje Update poruku svim ostalim cvorovima (kružno) da im javi o svom postojanju i svoje podatke.

-messageText (String) – predstavlja string portova razdvojenih zarezom. Na početku je prazna pa se redom dodaju port + “,”.

**QuitMessage**- Kada korisnik unese Quit komandu, radi se obrnuto od Update poruke gde se poruka salje svim ostalim cvorovima(kruzno) da uklone sve podatke o ovom cvoru. Takodje salje Bootstrapu poruku koja pocinje sa Bye i obavestava ga da ga ukloni iz aktivnih cvorova.

-originalSenderPort (int) – port originalnog posaljioaca, omogucava da poruka kruzno obidje sve cvorove.

-serventInfo (ServentInfo) - info serventa koji se iskljucuje

**StartMessage** – Kada korisnik unese Start komandu, moze je uneti na dva razlicita nacina:

- Ako korisnik unese samo Start bez argumenata, u komandnoj linije ce dobiti opcije za kreiranje novog posla gde korisnik treba da unese ime, broj tacaka, proporciju, duzinu i visinu pozadine kao i coordinate u parovima(x,y).
- Ako korisnik unese Start i naziv posla, proverava se da li je taj posao vec zadat(da li postoji u konfiguracionom fajlu) i ako da krece podelu za taj posao. Prvo se proverava koliko drugih poslova je zapoceto i koliko ima slobodnih cvorova za ovaj posao. Ako je neki drugi posao zapocet u isto vreme, sistem dodeljuje poslu odgovarajuci broj cvorova na kome radi. Kada dobije cvorove posao se deli na manje delove u odnosu na to koliko ima cvorova na raspolaganju. Delovi posla predstavljaju koordinate. Kada se posao podeli, delovi se dodaju u mapu pod kljucovima koji predstavljaju ID-jeve drugih cvorova. Ta mapa se salje kruzno kroz sve cvorove i oni proveravaju da li mapa sadrzi kljuc jednak njihovom ID-ju. Ako da, cvor poziva JobChaos koji izvorsava igru haosa za dodeljene coordinate. Po zavrsetku posla cvor salje kruznu poruku NewJobAdded kojom dodaje svoj rezultat u mapu rezultata za sve cvorove. Kada Start poruka dodje ponovo do originalnog posaljioaca takodje proverava i za njega da li ima posao za njegov ID i ako da izvorsava i dodaje rezultat.

- jobName (String) – ime posla
- originalSenderPort (int) – port originalnog posaljioaca, omogucava da poruka

kružno obidje sve cvorove.

- jobsForServentsByTheirIDs (Map<Integer, List<Point>>) – mapa gde ključevi predstavljaju ID-jeve cvorova a vrednosti njihove koordinate za posao

**NewJobAddedMessage** – Kada cvor izvrši svoj deo posla on šalje ovu poruku sa svojim rezultatom kružno, svim ostalim cvorovima i dodaje je u njihove structure za cuvanje rezultata.

- originalSenderPort (int) – port originalnog posaljioaca, omogućava da poruka kružno obidje sve cvorove.
- jobFractal (Job) – završen posao koji se dodaje u mape rezultata svih cvorova

**ResultMessage**- ovaj tip poruke se ne koristi zato što svaki cvor već ima sve rezultate od drugih cvorova pa nema potrebe da šalje poruku. Urađeno je na ovaj način da bi se ispostovalo pravilo za Map-Reduce gde se Reduce (sklapanje posla, ovde Result) izvršava na inicijatorskom cvoru.

**StatusMessage** – Kada korisnik unese Status komandu, moze je uneti na tri razlicita nacina:

- Ako korisnik unese samo Status bez argumenata, salje se poruka kruzno svim cvorovima gde bi oni trebalo da unosu svoje trenutno stanje izracunavanja.
  - Ako korisnik unese Status i naziv posla u poruku se dodaje taj naziv i trebalo bi da se vrate rezultati samo od cvorova koji rade taj posao.
  - Ako korisnik unese Status, naziv posla i fractal ID u poruku se dodaje naziv i fractalID i trebalo bi da se vrate rezultati samo od cvorova koji rade fractalID za taj posao.
- 
- jobname (String) – ime posla za koga trazimo status cvorova koji ga izvorsavaju
  - fractalID (int) – ID za koga trazimo status cvora koji ga izvorsava
  - originalSenderPort (int) – port originalnog posaljioca, omogucava da poruka kruzno obidje sve cvorove.
  - status (Map<Integer, Integer>) – mapa u koju ce cvorovi uneti svoj id i svoj status (velicina posla)