

Distribuirana igra haosa

KiDS projekat

Računarski fakultet

05.06.2020

Nevena Drešević

Email: ndresevic16@raf.rs

Indeks: RN 25/16

Grupa: 401

Uvod	2
Čvorovi u sistemu	2
Bootstrap	2
Konfiguracija bootstrap servera	3
Servent	3
Konfiguracija serventa	3
Arhitektura sistema	4
Uključivanje čvora u mrežu	4
Započinjanje novog posla	4
Raspodela posla	5
Zaustavljanje aktivnog posla	5
Isključivanje čvora iz mreže	5
Distribuirani mutex	5
Lamportov algoritam	6
Komande i izveštaji	6
status [X [id]]	6
start [X]	6
result X [id]	6
stop X	7
quit	7
Komunikacija između čvorova u mreži	7
Bootstrap i Servent	7
Servent i Servent	7
Rutiranje poruka u mreži	7
Tipovi poruka	8
NEW NODE	8
WELCOME	8
UPDATE	8
POISON	9
QUIT	9
JOB EXECUTION	9
ACK JOB EXECUTION	10
IDLE	10
ACK IDLE	11
COMPUTED POINTS	11
ASK STATUS	11
TELL STATUS	12



ASK JOB FRACTALID RESULT	12
TELL JOB FRACTALID RESULT	13
ASK JOB RESULT	13
TELL JOB RESULT	14
STOP JOB	14
REQUEST	14
REPLY	14
RELEASE	15
RELEASE CRITICAL SECTION	15

Uvod

Dokument predstavlja tehničku dokumentaciju za distribuirani sistem koji računa fraktalne strukture primenom [igre haosa](#). Prime 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.

U dokumentu je opisana arhitektura sistema, tipovi čvorova u mreži i njihovo korišćenje i ponašanje. Korisniku su predstavljene i komande koje služe za interakciju sa sistemom. Takođe, definisan je način komunikacije između čvorova sistema, protokol slanja poruka i one same.

Glavna uloga sistema je izvršavanje posla, tj. crtanja fraktalnih slika koja se ovde odvija distribuirano. Takođe, svi čvorovi u sistemu su jednaki i sistem je samoorganizujući.

Čvorovi u sistemu

U sistemu postoje dva tipa čvora, to su bootstrap (jedan u celom sistemu) i servent (čvorovi koji se uključuju i isključuju iz mreže).

Bootstrap

Predstavlja jedinstven čvor u sistemu koji služi isključivo za prvo uključivanje i isključivanje čvora iz mreže. Za funkcionisanje sistema neophodno je da bootstrap server bude aktivan u svakom trenutku i da ostali čvorovi imaju potrebne informacije kako bi ga kontaktirali (IP adresu i port).

Komunikacija sa bootstrap serverom svedena je na minimum i on nije svestan arhitekture sistema.

Konfiguracija bootstrap servera

Za pravilno poretanje bootstrap servera u sistemu, potrebno je obezbediti konfiguracionu datoteku koja sadrži sledeće parametre:

bootstrap.ip = ip adresa bootstrap servera

bootstrap.port = port bootstrap servera

Servent

Predstavlja tip čvora u mreži od kojih se sastoji sistem. Svaki servent je jedinstveno opisan svojom pozicijom u sistemu, IP adresom i portom. Servent ima mogućnost obavljanja zadatog posla i zajedno sa drugim serventima organizuje sistem. Takođe, ovaj tip čvora realizuje komunikaciju sa korisnikom preko konzole.

Konfiguracija serventa

Za pravilno poretanje čvora u sistemu, potrebno je obezbediti konfiguracionu datoteku i id (trenutno se ne koristi) kao ulazne parametre ServentMain klase. Konfiguraciona datoteka treba da bude sledećeg formata:

ip = ip adresa na kojem će čvor da sluša

port = port na kojem će čvor da sluša

bootstrap.ip = ip adresa bootstrap čvora

bootstrap.port = port bootstrap čvora

weak_failure_limit = slaba granica otkaza

strong_failure_limit = jaka granica otkaza

job_count = broj predefinisanih poslova

Svaki posao ima sledeće attribute:

job[index].name = naziv, jedinstveno simboličko ime za posao

job[index].points.count = broj tačaka fraktalne strukture. (int, $3 \leq N \leq 10$)

job[index].proportion = udaljenost između trenutne tačke i odredišta na kojoj će se pojaviti nova tačka (double u opsegu 0-1)

job[index].width = širina slike tj. površine na kojoj se računaju tačke (int)

job[index].height = visina slike tj. površine na kojoj se računaju tačke (int)

job[index].points.coordinates = skup koordinata tačaka u formatu (x,y) razdvojenih znakom “;”

Primer konfiguracije datoteke za serventa:

```
ip=localhost
port=1101
bootstrap.ip=localhost
bootstrap.port=2000
weak_failure_limit=1000
strong_failure_limit=10000
job_count=1
job0.name=triangle
job0.points.count=3
job0.proportion=0.5
job0.width=800
job0.height=800
job0.points.coordinates=(200,500);(500,200);(500,700)
```

Arhitektura sistema

Organizacija čvorova u sistemu konzistentna je u svakom trenutku rada sistema. Svakom čvoru u sistemu jedinstveno je dodeljen ID na osnovu kojeg se zna njegova pozicija u mreži. Arhitektura mreže predstavlja skup svih čvorova u sistemu numerisanih brojevima od **0** do **N - 1**, gde **N** predstavlja ukupan broj čvorova. Svaki čvor ima listu susednih čvorova sa kojima može direktno da komunicira, tj. ima vezu sa njima. U ovom sistemu svaki čvor ima vezu sa prvim sledbenikom, što predstavlja kružnu listu i dodatne veze, "skip" konekcije. Preciznije, **N** - ti čvor ima vezu sa $(N+2)^i$ - im (po modulu ukupnog broja čvorova u mreži), gde **i** uzima vrednosti iz intervala **[0..N]**.


Uključivanje čvora u mrežu

Kako bi se novi čvor uključio u mrežu njegov zadatak je da prvo kontaktira bootstrap server. Nakon odgovora bootstrapa koji sadrži podatke o poslednjem čvoru u sistemu, novi čvor mu se javlja slanjem NEW NODE poruke. Nakon toga, trenutni poslednji čvor uključuje novi čvor na kraj sistema, tj. dodeljuje mu ID za 1 veći od svog i pokreće ažuriranje čitavog sistema.

Započinjanje novog posla

Komanda start sa propratnim podacima vezanim za određeni posao služi za započinjanje novog posla u sistemu. Čvor nad kojim je iniciran ovaj zahtev tada zaustavlja sistem kako bi napravio novu organizaciju posla u sistemu i pravilo preraspodelio do sada urađeni posao kako ne bi došlo do gubitka podatka.

Raspodela posla



Pre raspodele čvorova unutar jednog posla, sistem deli svakom poslu čvorove koje može da koristi za svoje izračunavanje. Ova podela je jednaka koliko to dozvoljava ukupan broj aktivnih poslova i čvorova u sistemu. Zatim se čvorovi dele unutar jednog posla, ukoliko je broj dodeljenih čvorova određenom poslu takav da posao ne može da se podeli na delove gde svaki čvor ima svoj fraktalni region, tada postoje i idle čvorovi. Čvorovi koji nisu idle izvršavaju posao kreiranja fraktalne slike. Ti čvorovi imaju jasno definisan fraktalni ID, a time i deo slike koji računaju.

Pored novouspostavljene raspodele posla i čvorova u sistemu, ukoliko ovo nije inicijalna raspodela sistem ažurira, tj. održava da prethodno izračunate tačke i dalje pripadaju odgovarajućim poslovima i fraktalnim ID-jevima. Ovo se realizuje razmenom podataka između čvorova nakon dobijene nove raspodele.

Zaustavljanje aktivnog posla

Zaustavljanje aktivnog posla u sistemu uzrokuje slične događaje kao i započinjanje novog posla, tj. dovodi do reorganizacije i razmene podataka. Komanda *stop* sa parametrom koji predstavlja ime posla dovodi do privremenog zaustavljanja sistema i preraspodele. Čvor nad kojim je inicirana ova komanda zadužen je za restruktuiranje i slanje kružne poruke svim čvorovima u sistemu kako bi ih obavestio o novoj raspodeli. Takođe, čvorovi brišu sve izračunate tačke i podatke vezane za posao koji se zaustavlja.

Isključivanje čvora iz mreže

Isključivanje čvora iz sistema se realizuje komandom *quit*. Tada čvor koji napušta sistem započne kružnu poruku koja obavesti ostale čvorove o njegovom isključenju. Ta poruka ima ulogu ažuriranja globalnog stanja sistema i lista suseda. Nakon toga, opet se dešava reorganizacija izvršavanja poslova u sistemu.

Distribuirani mutex

Međusobno isključivanje je svojstvo koje je potrebno obezbediti u distribuiranim sistemima kako ne bi došlo do konkurentnosti među procesima. Uslov je da proces ne može da uđe u kritičnu sekciju dok je drugi proces trenutno u njoj. To znači da je samo jednom procesu dozvoljeno da izvrši kritičnu sekciju u bilo kom datom trenutku.

Zbog distribuirane prirode ovog sistema nemamo zajedničku memoriju niti zajedničko vreme, pa se problem ne može rešiti korišćenjem zajedničkih promenljivih. Stoga se koristi pristup zasnovan na prenošenju poruka i Lamportov algoritam.

Lamportov algoritam

Poruke u sistemu vezane za ovaj algoritam su REQUEST, REPLY i RELEASE i one poštuju FIFO vid komunikacije. Ova komunikacija u sistemu je realizovana na posebnom portu za svaki čvor. Osnovna ideja algoritma je se šalje REQUEST svim ostalim čvorovima kako bi se zatražila kritična sekcija. Zatim, čvor dobija REPLY poruku od svih ostalih čvorova i izvršava kritičnu sekciju. Nakon toga, oslobađanje kritične sekcije podrazumeva obaveštavanje čvorova o tome slanjem RELEASE poruke.

Takođe, svaki čvor ima podatke o zatraženim kritičnim sekcijama od ostalih čvorova koje su poredene po vremenu zahteva. Vremenska oznaka se dodeljuje se svakoj kritičnoj sekciji korišćenjem Lamportovog logičkog sata. Ta vremenska oznaka se koristi i za određivanje prioriteta zahteva kritičnih sekcija.

Komande i izveštaji

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 rade 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 je simboličko ime nekog posla navedenog u konfiguracionoj datoteci. Ako se X izostavi, korisnik unese parametre za posao na konzoli.

result X [id]

Prikazuje rezultate za završeno izračunavanje za posao X. Korisnik može, a ne mora da navede fraktalni ID za rezultat. Ako se zostavi, onda se dohvata rezultat za ceo posao, u suprotnom samo za taj fraktalni ID. Slika se eksportuje kao PNG fajl pod nazivom *"fractals/jobName_proportion.png"* ili *"fractals/jobNamefractalID_proportion.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.

Komunikacija između čvorova u mreži

Bootstrap i Servent

Komunikacija između bootstrap-a i serventa dešava se samo pri uključivanju i isključivanju serventa iz sistema. Bootstrap je u mogućnosti da primi i obradi sledeće tri vrste poruka:

- **Hail\nipX:portX\n** - poruka koju šalje servent koji želi sa se uključi u sistem. Ukoliko je on prvi čvor u sistemu, bootstrap odgovara sa **-1\n-1\n** dajući mu do znanja da je on prvi i da je uspešno uključen. Ako to nije slučaj, bootstrap odgovara porukom **ipX:portX\nipY:portY\n** koja sadrži informacije o odredištu prvog i poslednjeg čvora u sistemu.
- **New\nipX:portX\n** - poruka koju šalje servent nakon uspešnog uključivanja u sistem, bootstrap ga zatim dodaje u listu aktivnih servenata.
- **Quit\nipX:portX\n** - poruka koju šalje servent koji želi da se isključi iz sistema, bootstrap ga zatim izbacuje iz svoje liste aktivnih servenata.

Servent i Servent

Komunikacija između dva serventa se obavlja preko socket-a. Svaki čvor jedinstveno je određen svojom IP adresom i portom na kojem osluškuje poruke.

Poruka koja se razmenjuje između ovih čvorova predstavlja proširenje na *BasicMessage* poruke. *BasicMessage* je Java serijalizovana poruka koja sadrži IP adresu i port pošaljioca i primaoca, jedinstveni identifikator, pređenu putanju kroz mrežu do dostizanja krajnjeg primaoca, tekst poruke i druge attribute.

Rutiranje poruka u mreži

Poruke kroz mrežu mogu da se šalju direktno ukoliko su čvorovi susedi ili preko dozvoljenih skokova dobijenih na osnovu liste suseda svakog čvora. Određene vrste poruka je potrebno da pročitaju svi čvorovi u sistemu, pa se one šalju kružno kroz čitav sistem. Takve poruke se prosleđuju prvom susedu, sve dok poruka ne obiđe sve čvorove u mreži. Druge vrste poruka su namenjene specifično jednom čvoru, te se one rutiraju kroz mrežu dok ne dođu do svog odredišta. Zbog načina rutiranja i arhitekture sistema celokupna putanja poruke od čvora A do čvora B teži logaritamskoj zavisnosti od ukupnog broja čvorova u

sistemu. Ukoliko čvor A nije direktno povezan sa čvorom B, određuju se posredni čvorovi preko koji se prenosi poruka tako što se konstantno biraju čvorovi koji su najbliži čvoru B, a imaju direktnu vezu sa čvorom A, sve dok se ne dođe do čvora B.

Tipovi poruka

NEW NODE

Slanje poruke: Poruka koju šalje novi čvor nakon komunikacije sa bootstrap serverom, tj. nakon dobijenih informacija kome da se obrati. Šalje trenutnom poslednjem čvoru u mreži da želi da se uključi u sistem.

Prijem poruke: Pošiljaocu poruke dodeljuje ID za 1 veći od svog i obaveštava ga slanjem WELCOME poruke. Takođe radi zaključavanje sistema.

WELCOME

Slanje poruke: Poruka koju šalje čvor koji je uključio nov čvor u sistem.

Prijem poruke: Dobija svoj ID, inicijalizuje sve potrebne strukture za funkcionisanje u sistemu i svom susedu šalje poruku UPDATE zajedno sa svojim novim informacijama.

Dodatni sadržaj:

`int id` - id čvora koji se uključio

`String firstServentIpAddressPort` - odredište ka prvom čvoru u mreži, tj. prvom susedu čvora koji se uključio

UPDATE

Slanje poruke: Slanje kružne poruke koju započinje čvor koji se uključio u sistem.

Prijem poruke: Ažurira svoje podatke o ostalim čvorovima na osnovu dobijenih podataka iz poruke i prosleđuje prvom susedu ukoliko do njega ta poruka još uvek nije stigla. Šalje zahtev za puštanje kritične sekcije i zahtev koji inicira novu raspodelu posla.

Dodatni sadržaj:

`Map<Integer, ServentInfo> nodesMap` - čuva informacije o trenutnim čvorovima u sistemu, ključ mape je ID čvora a vrednost informacije o njemu

`Map<Integer, FractalIdJob> serventJobsMap` - informacije o trenutno aktivnim poslovima u sistemu, serventima koji rade na tom poslu i njihovim dodeljenim fraktalnim ID-jevima

`List<Job> activeJobs` - lista trenutno aktivnih poslova u sistemu

POISON

Koristi se za zastavljanje rada niti FifoSendWorker-a

QUIT

Slanje poruke: Kružno slanje poruke koju inicira servernt koji napušta sistem, nakon primljene *quit* komande.

Prijem poruke: Ažuriranje tabela suseda i informacija o čvoru koji je napustio sistem. Ukoliko postoji čvorovi koji nisu obavešteni o napuštanju čvora iz sistema, poruka se prosleđuje dalje. U suprotnom, šalje se zahtev za reorganizaciju sistema bez jednog čvora i odrađeni posao čvora koji je napustio sistem se šalje odgovarajućem čvoru.

Dodatni sadržaj:

`int quitterId` - id servernta koji je napustio sistem

`String jobName` - ime posla na kome je radio servernt

`String fractalId` - fraktalni ID za posao na kome je radio

`List<Point> quitterComputedPoints` - izračunate tačke pre napuštanja sistema

JOB SCHEDULE

Slanje poruke: Poruka koja se šalje kao zahtev za preraspodelu posla u sistemu, na primer nakon uključivanja novog čvora.

Prijem poruke: Inicira celokupnu preraspodelu posla u sistemu.

Dodatni sadržaj:

`JobScheduleType scheduleType` - tip događaja koji je uzrok preraspodele posla, može biti: *JOB_ADDED, JOB_REMOVED, SERVENT_ADDED, SERVENT_REMOVED*

JOB EXECUTION

Slanje poruke: Poruku šalje čvor koji je kreirao novu organizaciju poslova u sistemu ili čvor koji je uradio odgovarajući deo podele posla.

Prijem poruke: Ažuriranje podataka o novoj raspodeli posla, slanje prethodno izračunatih tačaka odgovarajućem čvoru ukoliko je čvor radio na nekom poslu u prethodnoj raspodeli. Ukoliko je čvor dobio više od jednog fraktalnog ID-ja, pravi dodatnu raspodelu posla na osnovu te liste i dobijenih krajnjih tačaka i šalje odgovarajućim čvorovima zaduženim za taj posao. U suprotnom, čvor je dobio konkretan posao koji treba da izvršava i započinje kreiranje tačaka. Takođe, slanje ACK JOB EXECUTION poruke tj. potvrde o prijemu JOB EXECUTION poruke čvoru koji je napravio preraspodelu.

Dodatni sadržaj:

`List<String> fractalIds` - lista fraktalnih ID-jeva za posao i granične tačke

`List<Point> startPoints` - krajnje tačke fraktala ili dela fraktala koji se računa

`Job job` - posao za koji je čvor zadužen

`int level` - nivo deljenja fraktalnih ID-jeva, tj. dubina podele jednog posla

`Map<Integer, FractalIdJob> serventJobsMap` - informacije o trenutno aktivnim poslovima u sistemu, serventima koji rade na tom poslu i njihovim dodeljenim fraktalnim ID-jevima

`Map<FractalIdJob, FractalIdJob> mappedFractalsJobs` - struktura koja predstavlja mapiranja odgovarajućih novih i starih fraktalnih ID-jeva i njihovih poslova

`List<Job> activeJobs` - lista trenutno aktivnih poslova u sistemu

`JobScheduleType scheduleType` - tip događaja koji je uzrok preraspodele posla, može biti: *JOB_ADDED, JOB_REMOVED, SERVENT_ADDED, SERVENT_REMOVED*

`int jobSchedulerId` - id čvora koji je započeo preraspodelu posla

ACK JOB EXECUTION

Slanje poruke: Poruka koju šalje čvor koji je dobio novo izvršavanje posla. Tj. poruka se šalje čvoru koji obavlja reorganizaciju sistema kao potvrda prijema JOB EXECUTION poruke.

Prijem poruke: Ažuriranje informacija o čvorovima koji se čekaju da pošalju potvrdu o prijemu poruke, tj. inkrementiranje broja vraćenih ACK tipova poruka.

IDLE

Slanje poruke: Nakon izvršene organizacije posla u sistemu, čvor koji je izračunao novu raspodelu šalje ovaj tip poruke čvorovima koji ne rade ni na jednom poslu.

Prijem poruke: Ažuriranje podataka o novoj raspodeli posla, slanje prethodno izračunatih tačaka odgovarajućem čvoru ukoliko je čvor radio na nekom poslu u prethodnoj raspodeli. Takođe, slanje ACK IDLE poruke tj. potvrde o prijemu IDLE poruke čvoru koji je napravio preraspodelu.

Dodatni sadržaj:

`Map<Integer, FractalIdJob> serventJobsMap` - informacije o trenutno aktivnim poslovima u sistemu, serventima koji rade na tom poslu i njihovim dodeljenim fraktalnim ID-jevima

`Map<FractalIdJob, FractalIdJob> mappedFractalsJobs` - struktura koja predstavlja mapiranja odgovarajućih novih i starih fraktalnih ID-jeva i njihovih poslova

`List<Job> activeJobs` - lista trenutno aktivnih poslova u sistemu

`JobScheduleType scheduleType` - tip događaja koji je uzrok preraspodele posla, može biti: *JOB_ADDED, JOB_REMOVED, SERVENT_ADDED, SERVENT_REMOVED*

`int jobSchedulerId` - id čvora koji je započeo preraspodelu posla

ACK IDLE

Slanje poruke: Poruka koju šalje čvor koji je idle u novoj raspodeli posla u sistemu. Tj. poruka se šalje čvoru koji obavlja reorganizaciju sistema kao potvrda prijema idle informacije nakon primljene poruke IDLE.

Prijem poruke: Ažuriranje informacija o čvorovima koji se čekaju da pošalju potvrdu o prijemu poruke, tj. inkrementiranje broja vraćenih ACK tipova poruka.

COMPUTED POINTS

Slanje poruke: Poruka koju šalje čvor koji je računao tačke za određeni posao i fraktalni ID, ali je došlo do preraspodele poslova u sistemu, pa je potrebno da "preda" svoje podatke čvoru koji preuzima deo ili ceo njegov posao.

Prijem poruke: Preuzimanje izračunatih tačaka ili dela tačaka za trenutni posao.

Dodatni sadržaj:

`String jobName` - ime posla za koji su izračunate tačke

`String fractalId` - fraktalni ID za koji su izračunate tačke

`List<Point> computedPoints` - izračunate tačke

ASK STATUS

Slanje poruke: Poruka koju inicira čvor kome je zatraženo trenutno stanje sistema ili dela sistema komandom *status*. Ukoliko je zatraženo stanje izračunatih tačaka za određeni posao i fraktalni ID, poruka se šalje jednom odgovarajućem čvoru, ako je zatraženo stanje za samo određeni posao, poruka se šalje prvom čvoru u sistemu koji radi na tom poslu. U suprotnom, ako se traži stanje celokupnog sistema, poruka se prosleđuje prvom susedu.

Prijem poruke: Ako je zatraženo stanje za određeni posao i fraktalni ID, po prijemu poruke čvor šalje svoje broj izračunatih tačaka u vidu TELL STATUS poruke serventu koji je prvobitno inicirao zahtev za status. Ukoliko je zatraženo stanje za samo određeni posao, čvor ispravno dodaje svoj broj izračunatih tačaka i prosleđuje dalje poruku čvoru koji radi taj posao, ako on nije poslednji na poslu. U suprotnom, ako je čvor poslednji, šalje TELL STATUS poruku inicijatoru. Treća verzija statusa predstavlja traženje za ceo sistem i ukoliko je ona aktivirana, čvor dodaje svoje stanje i prosleđuje poruku dalje ako postoje čvorovi koji

nisu dali informacije o svom stanju. Ako su svi čvorovi dali podatke o svom trenutnom stanju, analogno se šalje TELL STATUS poruka.

Dodatni sadržaj:

`String jobName` - ime posla za koji se traži status

`String fractalId` - fraktalni ID za koji se traži status

`Map<String, Map<String, Integer>> resultMap` - mapa koja za svaki aktivan posao u sistemu ima informacije o postojećim fraktalnim ID-jevima na tom poslu i broju nacrtanih tačaka na njima

`int version` - verzija zatražene *status* komande, može biti 0 (*status jobName fractalId*), 1 (*status jobName*) ili 2 (*status*)

TELL STATUS

Slanje poruke: Poruka koju šalje čvor koji je poslenji dodao svoje informacije o statusu, u zavisnosti od verzije zadate komande.

Prijem poruke: Oslobođanje kritične sekcije i štampanje prikupljenog statusa sistema korisiku.

Dodatni sadržaj:

`Map<String, Map<String, Integer>> resultMap` - mapa koja za svaki aktivan posao u sistemu ima informacije o postojećim fraktalnim ID-jevima na tom poslu i broju nacrtanih tačaka na njima

`int version` - verzija zatražene *status* komande, može biti 0 (*status jobName fractalId*), 1 (*status jobName*) ili 2 (*status*)

ASK JOB FRACTALID RESULT

Slanje poruke: Poruka koju šalje čvor koji je primio komandu *result* sa imenom posla i fraktalnim ID-jem za koji se traže izračunate tačke. Poruka se čvoru koji radi na tom poslu i ima traženi fraktalni ID.

Prijem poruke: Slanje TELL JOB FRACTAL RESULT poruke onom serventu koji je prvobitno inicirao zahtev za rezultat.

Dodatni sadržaj:

`String jobName` - ime posla za koji se traži rezultat

TELL JOB FRACTALID RESULT

Slanje poruke: Poruku šalje čvor koji radi na traženom poslu i ima fraktalni ID čiji se rezultat tražio.

Prijem poruke: Poruku prima čvor koji je dobio komandu *result* za traženi posao i fraktalni ID. Čvor oslobaša kritičnu sekciju, zatim prikazuje sliku kao rezultat prikupljenih tačaka i čuva je u fajlu pod nazivom "fractals/ jobNamefractalID_proportion.png".

Dodatni sadržaj:

String *jobName* - ime posla za koji se traži rezultat

String *fractalId* - fraktalni ID za koji se traži rezultat

List<Point> *computedPoints* - izračunate tačke za taj posao i fraktalni ID

int *width* - širina slike, tj. površine izračunavanja

int *height* - visina slike, tj. površine izračunavanja

double *proportion* - proporcija posla

ASK JOB RESULT

Slanje poruke: Poruka čije slanje započinje onaj čvor koji je primio komandu *result* sa imenom posla za koji se traži fraktal. Poruka se šalje prvom čvoru u sistemu koji radi na ovom poslu, zatim se prosleđuje do poslednjeg.

Prijem poruke: Dodavanje izračunatih tačaka na one tačke primljene iz poruke i dalje prosleđivanje poruke ukoliko postoji još čvorova koji rade na tom poslu a nisu dodali svoj deo posla. U suprotnom, slanje TELL JOB RESULT poruke onom serverntu koji je prvobitno inicirao zahtev.

Dodatni sadržaj:

String *jobName* - ime posla za koji se traži rezultat

int *lastServentId* - id poslednjeg (najvećeg) čvora u sistemu koji izvršava taj posao

List<Point> *comuptedPoints* - izračunate tačke za taj posao

TELL JOB RESULT

Slanje poruke: Poruku šalje poslednji čvor u sistemu koji radi na poslu čiji se rezultat tražio.

Prijem poruke: Poruku prima čvor koji je dobio komandu *result* za traženi posao. Čvor oslobaša kritičnu sekciju, zatim prikazuje sliku kao rezultat prikupljenih tačaka za posao i čuva je u fajlu pod nazivom "fractals/ jobName_proportion.png"

Dodatni sadržaj:

`String jobName` - ime posla za koji se traži rezultat

`List<Point> computedPoints` - izračunate tačke za taj posao

`int width` - širina slike, tj. površine izračunavanja

`int height` - visina slike, tj. površine izračunavanja

`double proportion` - proporcija posla

STOP JOB

Slanje poruke: Kružna poruka čije slanje započinje onaj čvor u sistemu koji je primio komandu *stop* za određeni posao.

Prijem poruke: Izbacivanje posla iz sistema i ažuriranje podataka vezanih za poslove, ukoliko je čvor radio na tom poslu onda prestaje sa radom. Prosleđivanje poruke dalje ukoliko svi čvorovi nisu obavešteni o prestanku rada, u suprotnom, slanje zahteva za reorganizaciju poslova u sistemu.

Dodatni sadržaj:

`String jobName` - ime posla čije se izvršavanje zaustavlja

REQUEST

Slanje poruke: Poruku šalje čvor koji zahteva kritičnu sekciju svim ostalim čvorovima u sistemu.

Prijem poruke: Ažuriranje logičkog časovnika i stavljanje zahteva za kritičnom sekcijom red čekanja. Nakon toga, servent šalje REPLY poruku čvoru koji je poslao zahtev.


REPLY

Slanje poruke: Poruka se šalje kao odgovor na REQUEST poruku.

Prijem poruke: Ažuriranje logičkog časovnika i uklanjanje servenata iz liste servenata na koje se čeka odgovor.

RELEASE

Slanje poruke: Poruku šalje čvor koji oslobađa lock, tj. svoju kritičnu sekciju. Ona se šalje svim ostalim čvorovima u sistemu.



Prijem poruke: Ažuriranje logičkog časovnika i uklanjanje određene poruke zahteva iz reda čekanja.

RELEASE CRITICAL SECTION

Slanje poruke: Poruku šalje čvor kako bi obavestio onaj čvor u sistemu koji trenutno drži kritičnu sekciju da može da je oslobodi jer su neophodni poslovi završeni.

Prijem poruke: Oslobađanje kritične sekcije.