

KiDS Projekat: Distribuirane kraljice (potpun graf)

Dokumentacija

Pregled svih tipova poruka:

- JOIN
- JOIN_RESPONSE
- HELLO_NEIGHBOUR
- HELLO_RESPONSE
- RESULT
- JOB
- REQUEST_TOKEN
- TOKEN
- RESUME
- PAUSE
- REQUEST_STATUS
- STATUS
- LEAVING
- STEAL_REQUEST
- STEAL_RESPONSE

Napomena*: Sve poruke su serializovani Java objekti

JOIN:

Ovu poruku salje cvor Bootstrap-u kao inicialnu poruku kada se ukljucuje u sistem.

Sadrzaj poruke: informacije o cvoru posiljaocu (ip, port), kome se sposlala poruka (ip, port), tip poruke po kome se prepoznaje

Napomena*: (ip, port) su predstavljeni u vidu ServentInformation objekta u cijem slucaju je ip i port pored drugih najbitnija informacija. Takodje ce u svim narednim porukama da se podrazumevaju default (BasicMessage) podaci, a to su tip poruke, primalac(ServentInformation), posiljaoc(ServentInformation)

JOIN_RESPONSE:

Predstavlja odgovor na poruku JOIN sa strane bootstrap-a. Ovde se završava komunikacija sa bootstrapom

Sadrzaj poruke: informacije o svim cvorovima koji se nalaze u sistemu (njih ce cvor primalac da prihvati kao direktne komsije)

HELLO_NEIGHBOUR:

Predstavlja poruku koja se inicijalno salje svim komsijama prilikom ukljucivanja u sistem. Svaki cvor koj primi ovu poruku zabeležava cvora posiljoca kao komsiju u svojoj memoriji. Kasnije koristi te informacije(ip,port) kako bi njemu slao poruke. Odmah nakon prijema ove poruke cvor primalac salje posiljaocu svoj finalResult objekat koj sadrzi do tad sve sračunate matrice, zajedno sa statusima za svaki X problem distribuiranih kraljica.

HELLO_RESPONSE:

Predstavlja odgovor na HELLO_NEIGHBOUR

Sadrzaj poruke: finalRezult objekat

Cvor primalac preuzima finalRezult i uz pomoc njega update-uje svoj finalRezult sa podacima koji fale (posto ce mu svaki cvor poslati taj objekat i postoji mogucnost da nisu svi objekti uskladjeni tj da u nekom obj fali po neka matrica koju je posiljaoc primio dok je ova poruka bila u tranzitu)

RESULT:

Poruka koju salje cvor svim susedima nakon sto je izracunao sve sto je imao dodeljeno i nije mogao da vrsi dalje kradju poslova jer su ostali cvorovi svi ispod limit-a.

Sadrzaj poruke: AllResults objekat koj u sebi za odredjenu NxN matricu sadrzi sve moguće kombinacije(koje su njemu bile dodeljene ili koje je ukrao) rasporeda N kraljica tako da se ne napadaju

Prilikom prijema ove poruke finalResults objekat se azurira sa novodobijenim podacima

JOB:

Poruka koju salje cvor-indicator na kome se "start" komanda izvorsila svim svojim susedima.

Sadrzaj poruke: ArrayList<Job> jobChunk predstavlja parce posla za resavanje problema distribuiranih kraljica. Job sadrzi u sebi <int collumn>, <int row> i N velicinu matrice i br. Kraljica za koje treba naci kombinaciju gde je vec na matricu NxN na poziciju [collumn][row] postavljena jedna kraljica. Posao je tako raspodeljen da za matricu NxN ima NxN Job-ova koji se grupisu u chunk-ove i ravnomerno raspodele po svim cvorovima u sistemu. Ova granularnost Job-a bi bila pozeljna da se smanji u nekom buducem trenutku.

Cvor primalac istovaruje sve dobijene Job-ove u jobQueue(BlockingQueue) koj obradjuje komponenta "Calculator".

REQUEST_TOKEN: (Medjusobno iskljucivanje)

Poruka koja se salje kada Cvor zeli da pristupi kriticnoj sekciji. Posto samo cvor koj ima "token" ima pravo da pristupi kriticnoj sekciji ovim putem se salje zahtev za token.

Salje se svim susedima. Cvor koj primi ovu poruku zabelezava zahtev da cvor posiljaoc trazi token. Ukoliko cvor koj primi poruku ima "token" i nije u kriticnoj sekciji on prosledjuje cvoru posiljaocu token. Ukoliko je u kriticnoj sekciji posto je ovaj zahtev svakako zabelezen predace token dalje onom cvoru koj je prvi sledeci u redu za pristupanje kriticnoj sekciji nakon sto izadje iz kriticne sekcije. (kriticna sekcija se koristi kada cvor startuje posao ili zahteva status trenutnih proracuna)

TOKEN: (Medjusobno iskljucivanje)

Poruka koja se salje kada se prosledjuje dalje token cvoru koj ga je zahtevao tj. Cvoru koj zeli da pristupi kriticnoj sekciji

Sadrzaj poruke: Token

Salje se iskljucivo kada cvor nije u kriticnoj sekciji

PAUSE:

Salje se ukoliko je korisnik na cvoru zadao komandu "pause". Ili kada se startuje neki novi posao. Tada se salje PAUSE za trenutni aktivan posao. Salje svim svojim susedima.

Sadrzaj poruke: int N – parametar koj govori na koju NxN matricu tj. na koje Job-ove se odnosi

Cvor kada primi ovu poruku iz job queue-a izbacuje sve job-ove sa datim parametrom N i cuva ih u ArrayList<Job> pausedJobs, kako bi ih kasnije ukoliko se "start X" komanda sa odgovarajucim parametrom X zadala ponovo prebacili u jobQueue ili ukoliko dobiju poruku RSUME sa odgovarajucim parametrom. Markira u finalResult objektu za parametar N da je posao pod statusom "Paused".

RESUME:

Salje se ukoliko se "start X" komanda pokrenula za X parametar koj je vec zapocet da se racuna. Salje svim svojim susedima.

Sadrzaj poruke: int N – parametar koj ukazuje na to koji Job-ovi iz pausedJobs liste se prebacuju u JobQueue kako bi se dalje obradjivali

REQUEST_STATUS:

Salje se ukoliko se na cvoru zadala komanda "status". Salje svim svojim susedima.

STATUS:

Predstavlja odgovor na poruku REQUEST_STATUS. Salje se cvoru posiljaocu koj je posalo REQUEST_STATUS.

Sadrzaj poruke: ArrayList(SimpleImmutableEntry<Integer, String>> status

Status objekat predstavlja niz Entry-a sa kljucem Integer koj predstavlja Posao NxN matrice i value String koj predstavlja trenutni status tog posla. Cvor koj primi REQUEST_STATUS iz finalRezults objekta uzima status svih svojih poslova. Status posla moze da bude "Active", "Paused", "Done". Cvor primalac koj je poslo REQUEST_STATUS kada dobije od svih svojih suseda STATUS poruku on poredi njihove i svoj status za svaki kljuc ukoliko se medju dva entry-a nadju nesuglasice status se markira kao "Fuzzy" i ispisuje korisniku.

Ovakva nesuglasica moze da se dogodi kada se prima REQUEST_STATUS na cvoru koj kao poslednji završava posao za odredjeni matricu, sagledava da je 100% gotovo sračunata i postavlja status za taj posao kao "Done" dok poruka o njegovom obavljenom poslu (RESULT) jos nije stigla do ostalih suseda kod kojih je posao markiran kao "Active".

LEAVING:

Salje se kada je na cvoru pokrenuta komanda "stop" i on napusta sistem. Salje svim svojim susedima. Koji onda njega brisu iz liste komsija (suseda). Takodje prvom svom susedu salje svoj jobQueue i token ukoliko ga ima kako se nista ne bi izgubilo iz sistema.

STEAL_REQUEST: (Work stealing)

Salje se ukoliko cvor nema posao ili je završio posao. Tj kada je jobQueue prazan. Nasumično cvor bira kome će da salje poruku, ukoliko primaoc poruke odgovori sa STEAL_REQUEST porukom koj je prazna tj nije mu dodelio nijedan posao onda nasumično bira dalje sledeći cvor kome će da salje STEAL_REQUEST tako da se prethodni cvorovi ne ponavljaju. Kada prođe kroz sve susede znaci da nema šta više da pokrade salje RESULT poruku svim svojim susedima.

Sadržaj poruke: int limit

Limit govori primaocu koliki je limit njegovog preostalog posla koj ima koj sme da preda cvoru posiljaocu. Uzima približno polovinu posla iz JobQueue i vraća nazad cvoru posiljaocu putem poruke STEAL_RESPONSE.

STEAL_RESPONSE: (Work stealing)

Salje cvoru koj mu je poslao STEAL_REQUEST zajedno sa polovinom svog posla iz jobQueue-a ukoliko nije ispod limita.

Sadržaj poruke: ArrayList<Job> stolenJobs

Primalac uzima stolenJobs i dodaje ih sve u svoj jobQueue.