IKP Projekat

Tim 13

Zadatak 2

Studenti:

Dušan Radulović PR36/2018

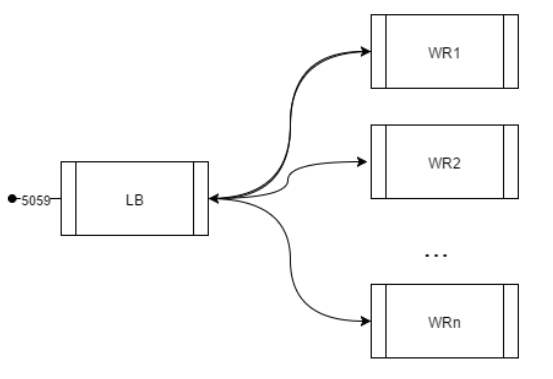
Karlo Pest PR57/2018

Asistent:

Marina Stanojević

Tema projekta: Load Balancer

Šta je Load Balancer?



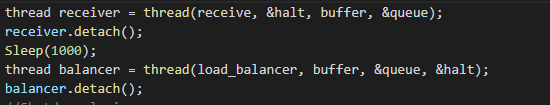
Load balacer je tip arhitekture u distribuiranim sistemima koji ima odgovornost da ravnomerno  
rasporedi posao koji je potreban da se izvrši na veći broj izvršilaca (worker-a) pri tom može imati i mogućnost da po potrebi poveća/smanji broj izvršilaca kada primeti da je to potrebno.

Realizacija Load Balancera u projektu

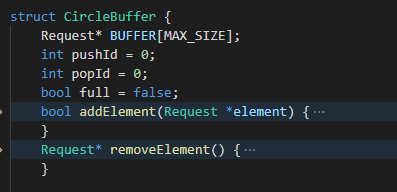
Put Diagram here

Klijenti se priključuju na sistem prilikom pokretanja, uspostavlja se TCP (Nonblocking) veza i šalje se poruka.

Na glavnom elementu sistem, Load Balancer-u se prilikom pokretanja kreiraju sva početna polja koja će biti korištena za rad i skladištenje poruka i podataka. Takođe se pokreće pozadinski proces Recieve i proces koji će izvršavati sam posao Load Balancer-a.



Oba ova procesa rukuju sa isitim buffer-om, ovaj buffer je kružni buffer.

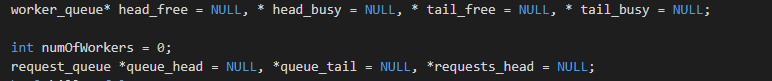
U ovaj buffer će se skladištiti podaci koji se prime od klijenta i iy njega će se uyimati i prosleđivati worker-ima.

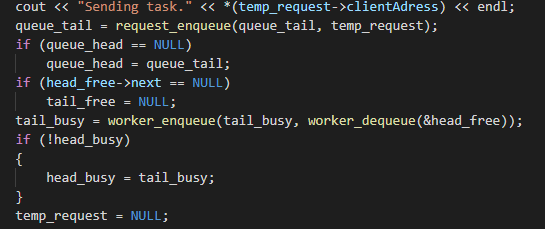
/\* Podaci koji se skladište su tipa Request\* , Request element u sebi sadrži SOCKET preko kojeg se komunicira sa klijentom i poruku koju je klijent poslao \*/

Glavi algoritam Load Balancer-a je sledeći:

1. Početni korak pre ulaska u petlju, inicijalizacija svih potrebnih promenljivih. Ovde se inicijalizuju liste kojima će se rukovati sa worker-ima.

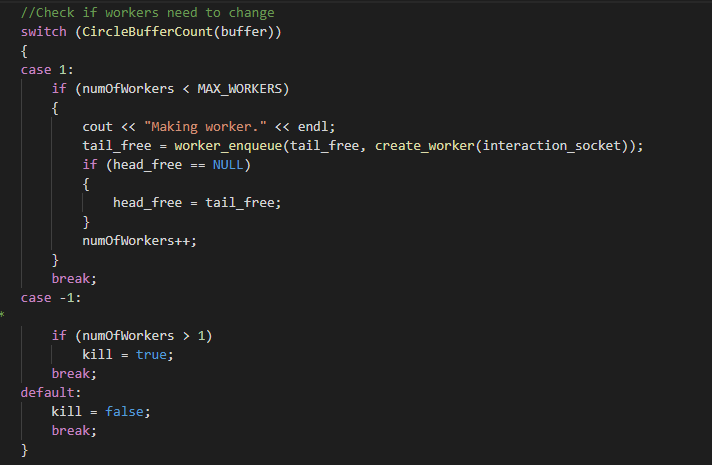
Ovde se takođe kreira neblokirajući Socket preko kojeg će Dispečer da prima poruke od worker-a kada oni završe sa obradom zahteva.

1. Prvi Korak petlje je provera da li ima zahteva za obraditi i da li ima slobodan worker.
   1. Ako ima zahteva koji se trebaju obraditi, Load Balancer nađe slobodnog worker-a, pošalje mu zatev. U ovom procesu se takođe worker skida sa liste slobodnih i ubacuje u listu zauzetih.



1.2. Ako nema zahteva za obradu ili slobodnog worker-a on preskaće ovaj korak.

2. Provera popunjenosti kružnog buffera



Ovde Load Balancer proverava da li treba da ugasi ili upali workera, Ako je kružni buffer popunjen preko 70%, kreira se novi worker ako nije već kreirano maximalan broj workera. Ako je nivo popunjenosti iznad 30% onda gasi sledećeg worker-a koji se javi da je slobodan.

1. Provera da li se worker javio da je završio obradu zahteva i ako jeste postavi taj zahtev u listu gotovih zahteva.
   1. U ovom delu algoritma se isto radi gašenje worker-a ako je to potrebno da se uradi

Proces koji radi u pozadini zajedno sa Load Balancerom je response process koji pristupa istoj listi zahteva kojoj pristupa i Load Balancer, to je lista gotovih zahteva. Response proces vrši jednostavan algoritam gde proverava da li ima gotovih procesa i ako ima vrati klijentu poruku pomoću SOCKET\* elementa koji se nalazi u samom zahtevu.

Stress testiranje

Ovaj program je testiran tako što je u veoma brzom momentu bilo upaljeno 50 instanci klijenata koji su, relativno, u isto vreme poslali zahtev. Pored ovog velikog broja klijenata, ovaj program je uspeo da odgovori svakom klijentu u prihvatljivo vreme i adekvatno je palio/gasio workere kada je bilo potrebno.

Zauzeće memorije od strane Load Balancer aplikacije je bilo konzistentno pre i posle izvršavanja rada klijenata, to jest nije bilo curenje memorije i zauzeće je bilo privatljivo.