

Dokumentacija projekta

Load Balancer

Predmet:

Industrijski komunikacioni protokoli u EES

Članovi tima:

Perišić Srđan PR 52/2021

Milanović Aleksandar PR 59/2021

Uvod

Opis problema koji se rešava

Load Balancer je sistem za upravljanje zahtevima između klijenata, radnika i komunikacije putem mreže. Ovaj sistem omogućava komunikaciju između klijenata, Load Balancera (LB) i radničkih (Worker) instanci koristeći socket protokol. Cilj projekta je omogućavanje pravilne raspodele zahteva između radnika uz optimizaciju i sinhronizaciju putem Load Balancer arhitekture.

Glavne komponente sistema

Load Balancer (LB): Prima zahteve od klijenata i raspoređuje ih među radnicima pomoću algoritma Round Robin.

Worker: Radni procesi koji obrađuju poruke primljene od LB.

Klijenti: Pošaljuju zahteve preko Load Balancera ka radnicima.

Komunikacija između sistema: Koristi se putem TCP protokola.

Ciljevi Zadatka

Implementirati Load Balancer kao glavnu serversku komponentu koja prima zahteve klijenata i prosleđuje ih workerima na obradu.

Omogućiti klijentu slanje zahteva.

Omogućiti workerima da čuvaju poruke, skladiste ih i repliciraju.

Dizajin

Client

Komponenta koja komunicira direktno sa Load Balancerom.

Glavna funkcionalnost:

Povezivanje sa LB: Klijent se povezuje sa LB na adresu localhost:5059.

Provera validnosti poruke: Proverava se da li poruke koje šalje klijent zadovoljavaju određene uslove pre slanja.

Slanje poruka ka LB.

Prijem odgovora od LB.

Glavne funkcije:

`send_message()`

`receive_message()`

`is_valid_message()`

Load Balancer

Glavni zadatak Load Balancera je primanje poruka od klijenata i njihovo prosleđivanje među radnicima (Worker) u sistemu korišćenjem Round Robin algoritma.

Glavna funkcionalnost:

Prihvatanje zahteva od klijenata.

Provera i ažuriranje svih Worker instanci.

Raspodela zahteva među Worker instancama po principu Round Robin algoritma.

Obrada poruka i održavanje strukture radnika.

Glavne funkcije:

AcceptConnectionsThread()

SendMessageToWorkersRoundRobin()

process_new_request()

send_message()

Worker

Worker procesi obrađuju poruke koje im Load Balancer prosleđuje.

Glavna funkcionalnost:

Dobija poruke od LB putem TCP komunikacije.

Čuva podatke u lokalnom skladištu.

Sinhronizuje sve podatke sa svim drugim radnicima u sistemu.

Glavne funkcionalnosti:

receive_combined_data()

process_message()

deserialize_combined_data_structure()

Common

Sadrži zajedničke funkcionalnosti kao što su:

Funkcije za sinhronizaciju (koristeći hash tabele).

Funkcije za upravljanje redovima poruka.

Proveru podataka, kao i obavljanje osnovnih funkcionalnosti kao dodatak za Worker, LB, i Client sisteme.

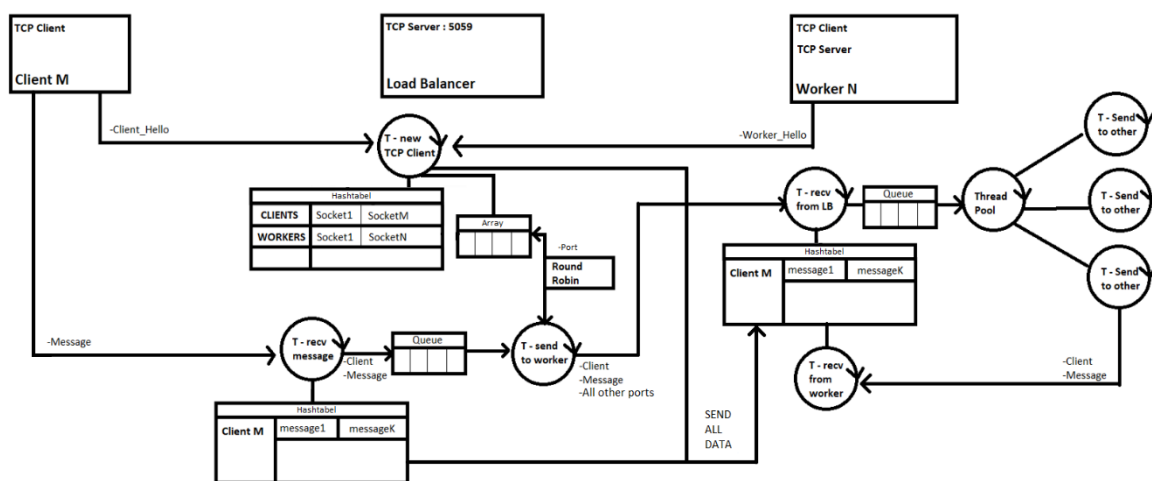
Glavne funkcionalnosti:

init_hash_table()

serialize_message()

add_table_item()

Arhitektonski dijagram



Tehnička objašnjenja

Load Balancer

Algoritam Round Robin:

Poruke od klijenata se prosleđuju Worker instancama po redosledu u kojem su radnici inicijalizovani, uz pomeranje indeksa svakim novim zahtevom.

Worker

Worker obavlja skladištenje podataka i sinhronizaciju podataka između ostalih radnika. Sve poruke koje Worker primi automatski se repliciraju među svim radnicima.

Glavne funkcionalnosti

Klijent šalje poruku Load Balanceru:

Proverava validnost podataka.

Ako validno, šalje poruku LB-u.

Load Balancer prosleđuje zahteve Workerima po principu Round Robin.

Worker preuzima poruku, čuva lokalno, i sinhronizuje je sa ostalim radnicima.

Sinhronizacija podataka među Worker instancama.

Radnici komuniciraju i ažuriraju svoj lokalni prostor kako bi svi imali iste podatke.

Primer upotrebe

Scenario:

Klijent šalje poruku "data1" ka LB.

LB preusmerava zahtev Worker-ima koristeći algoritam Round Robin.

Worker prima poruku, čuva podatke, i šalje potvrdu LB-u.

Ostali Worker-i sinhronizuju svoj prostor podataka.

Strukture Podataka

Korišćene su samostalno implementirane structure podataka:

Hash table za skladištenje poruka poslatih od klijenata, i za skladištenje socketa.

Queue kao struktura koja je implementiranao kako bi čuvala podatke koje treba obraditi da bi se smanjile nepravilnosti prilikom sinhronizacije.

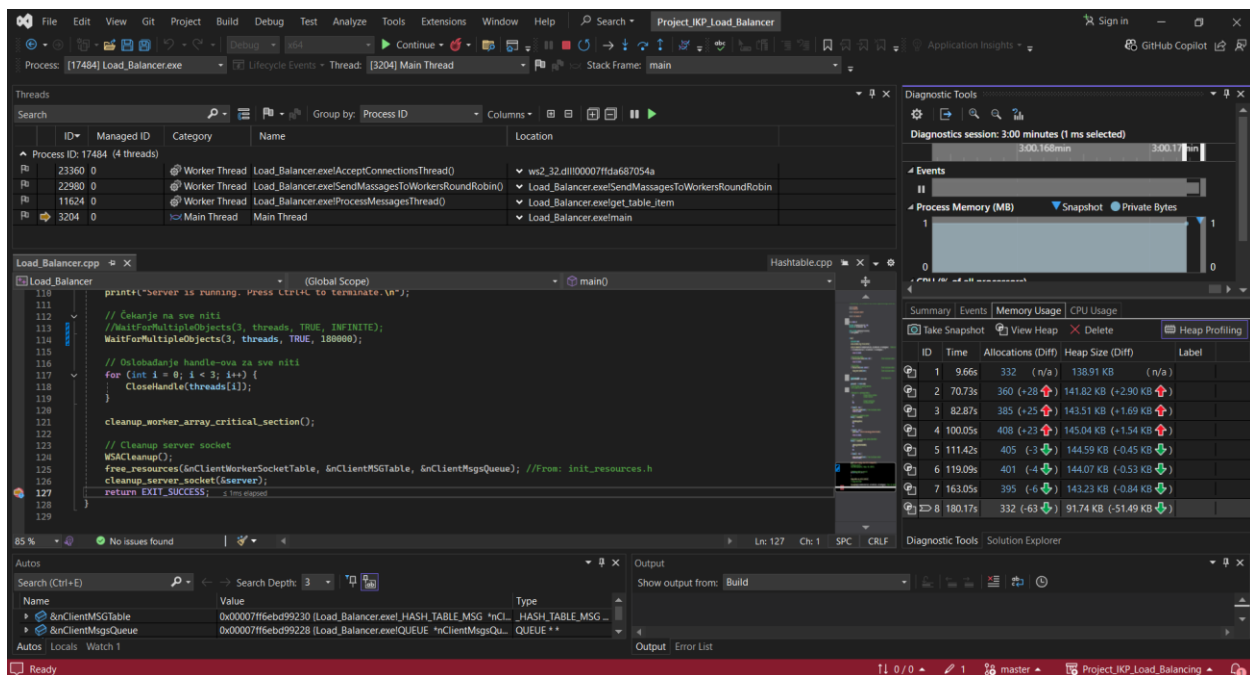
Rezultati Testiranja

Opis:

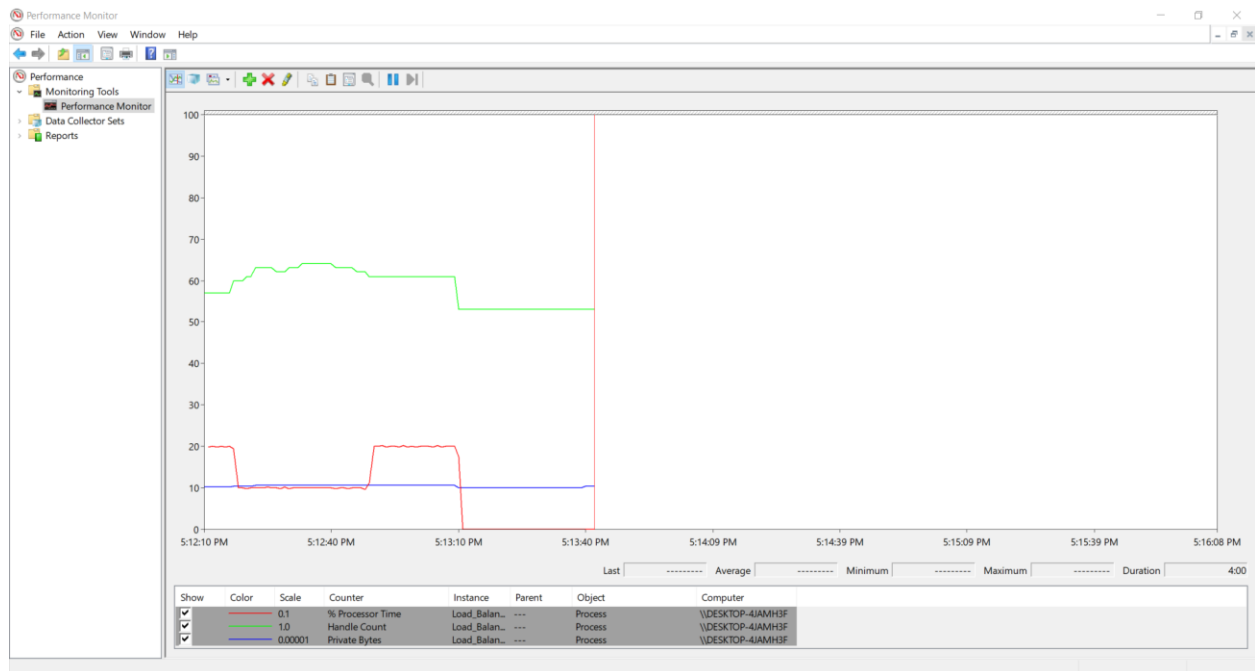
- Testiranje heap memorije, njene alokacije i čišćenja nakon završetka rada programa (UMDH)
- Performance counter testiranje

Prikazani rezultati testova:

Testiranje memorije:



Performance monitor:



Zaključak:

Rezultati testiranja memorije pokazuju da se skoro svi resursi koji su zauzeti nakon završetka rada programa očiste što znači da je curenje minimalno, odnosno da su rezultati dobri.

Potencijalna unapređenja

Dodavanje sigurnosnih mehanizama(enkripcija, autentifikacija)

Implementacija logovanja I praćenja aktivnosti

Implementacija mehanizama prioriteta poruka

Implementacija mehanizama odgovora

Tehnička specifikacija

Protokol:

TCP/IP protokol za komunikaciju.

Održavanje

Prilikom dodavanja novih radnika u sistem, LB automatski sinhronizuje sve radnike.

Redosled operacija se obavlja uz sinhronizaciju korišćenjem CRITICAL_SECTION za zaštitu deljenih resursa.