

Industrijski komunikacioni protokoli u elektroenergetskim sistemima - LoadBalancer

Srđan Mlađenović PR72/2020

Nikola Inđić PR150/2020

Zadatok projekta je implementacija Loadbalancer komponente u

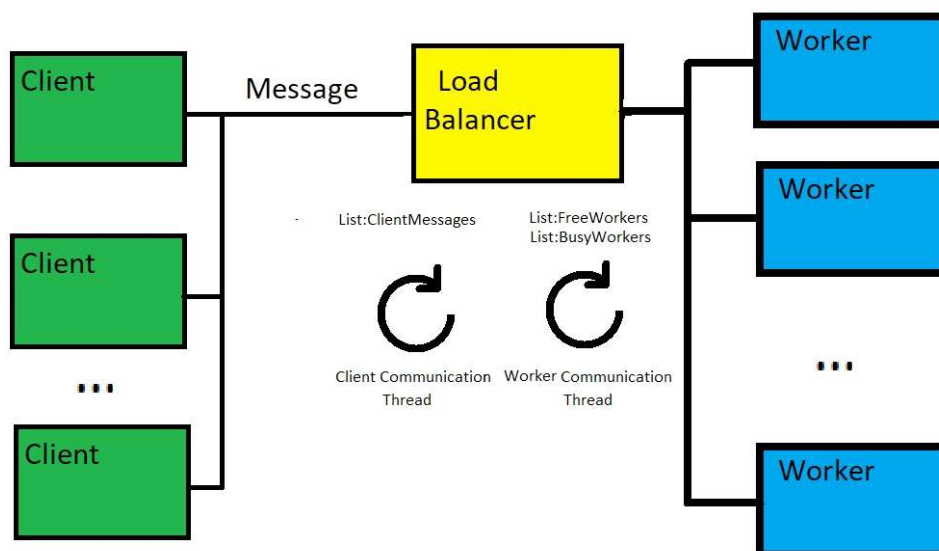
jednostavnoj server-klijent baziranoj aplikaciji.

Loadbalancer je komponenta koja ima zadatak da na određenom principu rada vrši ravnomernu raspodelu podataka kako bi se poboljšale preformanse same aplikacije.

U ovom slučaju zadatak Loadbalancer-a je da na određenom portu sluša zahteve klijenata, koji nakon konekcije mogu da šalju zahteve za obradu.

Loadbalancer potom skladišti zahteve i zatim u zavisnosti od broja randika kojima raspolaže ravnomerno raspoređuje zahteve za obradu koje radnici izvršavaju i nakon obrade javljaju da su završili posao.

Blok dijagram i opis rada programa:



Aplikacija se sastoji iz tri komponente: Loadbalancer (server), Client (client) i Worker (client).

Loadbalancer komponenta predstavlja server za dve klijentske, Client i Worker komponente. Server sluša na dva odvojena porta i uz pomoć threadova koji se izvršavaju istovremeno održava konekciju sa više instanca dve komponente u istom trenutku.

Kada neki od klijenata uspostavi konekciju ima mogućnost da pošalje zahtev za obradu. Server potom taj zahtev skladišti u listu klijentskih zahteva, zatim na osnovu informacija koje ima o radnicima iz liste slobodnih radnika on bira radnika koji ima najmanje posla i zatim šalje taj zahtev worker komponenti na obradu. Zatim se worker šalje u listu zauzetih radnika dok ne obradi zahtev i potom javlja serveru da je završio. Kada server dobije poruku da je worker završio ponovo obnavlja listu slobodnih i zauzetih workera. Kada worker uspostavi konekciju sa serverom, server iz dodaje u raspoložive radnike, kada izgubi konekciju on se briše iz raspoloživih radnika i ukoliko ima nedovršenih zahteva za obradu, oni se vraćaju serveru koji ih potom ponovo ravnomerno raspoređuje na slobodne workere u tom trenutku. Na taj način se vodi računa o ravnomernom raspoređivanju posla između radnika u svakom trenutku.

Tehnologije:

Program je realizovan u potpunosti u C programskom jeziku, uz biblioteke:

- Windows.h
- Winsock2.h

Strukture:

- list

Potencijalni razvoj:

- Fokus aplikacije bilo je samo raspoređivanje zahteva, koji su zapravo samo obični stringovi. U daljem razvoju, ti zahtevi mogu biti složeniji kako bi izvršali nešto konkretnije.
- Potencijalno bolje iskorišćenje memorije uz pomoć naprednijih

struktura podataka u odnosu na listu.