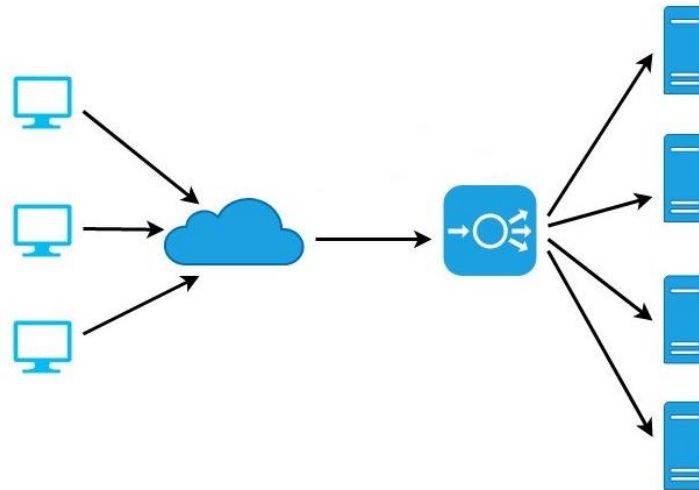


### Zadatak 1 – 19 poena

Potrebno je razviti skalabilnu arhitekturu za izvršavanje proračuna. Arhitektura se sastoji od klijenta koji šalje zahtev za proračunom, zatim servisa za balansiranje opterećenja i više radnika koji izvršavaju proračune.



#### Klijent

Klijent kreira zahteve za proračunom. Zahtev sadrži dva slučajna cela broja. Komunikacija između klijenta i balansera opterećenja vrši se pomoću stream soketa. Očekuje se jedan od dva moguća odgovora

- odbijanje zahteva
- rezultat proračuna.

Nakon dobijenog odgovora klijent ispisuje odgovor u konzolu i zatim pravili pauzu u trajanju od 5 do 10 sekundi, a potom startuje novi zahtev.

#### Balanser opterećenja

Predstavlja glavnu komponentu ovakvog sistema. Sadrži listu dostupnih radnika i informacije da li trenutno obrađuju zahtev ili su slobodni, kao i broj već obrađenih zahteva za svakog radnika. Servis može obraditi 3 vrste zahteva:

- **proračun:**
  - ako ne postoji ni jedan registrovani radnik, zahtev se odbija
  - zahtev se raspoređuje na slobodnog radnika koji je do sada obradio najmanji broj zahteva, a ukoliko nema slobodnog radnika, čeka do trenutka kada neki radnik postane slobodan i potom se raspoređuje.
- **registracija novog radnika**
  - prihvataju se podaci o adresi i portu RMI servisa radnika. Nakon ažuriranja liste radnika, novi radnik se može upotrebiti za izvršavanje proračuna.
- **odjavljivanje registrovanog radnika.**

- prihvataju se podaci o adresi i portu RMI servisa radnika. Ako se servis nalazi u listi radnika, servis se odjavljuje. Ukoliko pomenuti servis trenutno izvršava neki proračun, odjava se vrši tek po završetku proračuna.

Svaki od navedenih zahteva obrađuje se u zasebnoj niti.

### Radnik

Zaduženje radnika je izvršavanje proračuna. Predstava RMI servis jednom metodom za izvršavanje proračuna. Metoda prihvata dva cela broja. Simulirati trajanje proračuna u trajanju od 3 do 15 sekundi. Rezultat proračuna je proizvod prosleđenih brojeva.

Na startu servisa, radnik odmah vrši prijava na prethodni servis. Sve vreme može se sa konzole uneti naredba STOP nakon koje se šalje zahtev za odjavljivanje. Nakon uspešne odjave radnik se zaustavlja.

### Ograničenja

Zahtevi koje klijent upućuje balanseru opterećenja koriste stream sokete – definisati pogodni protokol za komunikaciju. Isto pravilo važi i za odgovore na zahteve klijenata. Zahtevi za prijavu i odjavu servisa radnika takođe koriste stream sokete, ali uz obaveznu upotrebnu XML-a. Komunikacija ka radnicima odvija se upotrebom RMI protokola.

### Zadatak 2 – 4 poena

Napraviti servis za dobijanje tačnog vremena upotrebom UDP protokola. Server na svakih 5 sekundi šalje podatke o trenutnom vremenu. Klijent u konzoli ispisuje podatke o tačnom vremenu u trenutku kada podaci stignu.