RASSUS odgovori na pitanja iz prezentacije i knjige (2. predavanje)

22. studenoga 2022.

1. **Objasnite zašto tranzijentna sinkrona komunikacija potencijalno pati od problema vezanih uz skalabilnost.**

   Uzmimo na primjer da imamo jednog korisnika. On pošalje poruku poslužitelju i bududi da je komunikacija sinkrona korisnik je blokiran, ne radi ništa dok ne primi odgovor. Ali, bududi da je komunikacija tranzijentna mogude je da poslužitelj nije dobio poruku i korisnik tada ulazi u beskonačnu petlju čekanja. Pri povedanjem skalabilnosti se povedava broj korisnika kojima sustav ne radi ispravno.

2. **Može li se pomoću UDPa implementirati protokol za pouzdanu komunikaciju između klijenta i poslužitelja? Ako može, na koji način?**

   Može se implementirati pomoću slanja potvrda za svaki UDP paket. Ako korisnik 1 nije dobio potvrdu od korisnika 2 za primitak poslanog UDP paketa nakon određenog vremena, korisnik 1 isponova šalje taj određeni UDP paket.

3. **Poslužitelj je implementiran pomoću socketa TCP na portu 10000 s ograničenjem `NUMBER_OF_THREADS`=2. Objasnite detaljno operacije prilikom dolaska prvog klijentskog zahtjeva na poslužitelj. Što se događa kada stigne drugi, pa treći klijentski zahtjev, a prve dvije konekcije su još uvijek aktivne. Koliko socketa je vezano uz port 10000?**

   JOŠ SE MORA NAĆI ODGOVOR!

4. **Koliko byte-a se može maksimalno zapisati u UDP datagram?**

   65 515 bajtova (uključujući 8 bajtova UDP zaglavlja)*

5. **Objasnite na primjeru razliku između perzistentne i tranzijentne komunikacije.**

   Primjer perzistentne komunikacije bi bio komunikacija porukama. Korisnik 1 šalje poruku korisniku 2 i sustav jamči da de korisnik 2 dobiti tu poruku, bez obzira je li mu upaljen ili ugašen uređaj. Primjer tranzijentne komunikacije je chat sustav baziran na UDP-u gdje korisnik 1 šalje poruku korisniku 2, ali se ne jamči isporuka poruke ako korisnik 2 nije aktivan.

6. **Objasnite razliku između sinkrone i asinkrone komunikacije.**

   Kod sinkrone komunikacije pošiljatelj je blokiran nakon slanja poruke sve do primitka potvrde o isporuci ili sve do primitka odgovora od poslužitelja, ovisno o implementaciji. Kod asinkrone komunikacije pošiljatelj nije blokiran te nastavlja procesiranje odmah nakon slanja.

7. **Navedite prednosti koje ima operacija slanja zahtjeva koja je neblokirajuća u odnosu na blokirajuću operaciju.**

Korisnik koji šalje zahtjev nije blokiran i može nastaviti s radom dok čeka zahtjev. Za to vrijeme može poslati nove zahtjeve drugim poslužiteljima ili može napraviti neki unutarnji posao.

8. Ima li smisla ograničiti broj dretvi za obradu korisničkih zahtjeva na višedretvenom poslužitelju? Zašto?

Valjda ima smisla jer se ostatak dretvi koristi za npr. obradu unutarnjih poslova.

9. Je li poslužitelj koji održava TCP/IP konekciju prema klijentu stateful ili stateless?

Poslužitelj je stateful jer postoje zahtjevi unutar TCP/IP protokola koji mijenjaju stanje konekcije, npr. ako se korisnik želi odspojiti, poslati će zahtjev za raskid konekcije.

10. Navedite prednosti konkurentnog poslužitelja u odnosu na iterativni poslužitelj.

Prednost je što konkurentni poslužitelj može posluživati više korisnika odjednom i samim time se povećava efikasnost sustava.

11. Usporedite gRPC i RESTful servise u pogledu performanci.

gRPC koristi *protobuf* koji smanjuje veličinu poruke, što ga čini bržim od REST servisa. Dok rade s HTTP 2, mikroservisi primaju višestruke klijentske zahtjeve i postižu multipleksiranje istodobnim posluživanjem različitih zahtjeva i odgovora koristeći podatke na jednoj TCP vezi. Stoga gRPC eliminira ograničenja REST API-ja njihovim kapacitetom stalnog protoka podataka.Kada REST mikroservisi primaju više zahtjeva od više klijenata istovremeno, oni bivaju usluženi, a time se cijeli sustav usporava. Također, kada se više zahtjeva primi paralelno, nije lako vratiti odgovor klijentu istim redoslijedom