

1 Uvod u raspodijeljene sustave

1.1. Objasnite ulogu programskog posredničkog sloja za raspodijeljene sustave.

- Programski posrednički sloj omogućava povezivanje, komunikaciju i suradnju aplikacija, sustava i uređaja te omogućuje interakciju programa na aplikacijskoj razini. Nalazi se između operacijskog sustava i aplikacijskih programa u programskoj arhitekturi sustava. Cilj je olakšati korisnicima i aplikacijama pristup i korištenje udaljenih sredstava na kontroliran i učinkovit način.

1.2. Usporedite migracijsku i relokacijsku transparentnost raspodijeljenog sustava. Koje svojstvo (ne)zadovoljava web poslužitelj i zašto?

- Migracijska transparentnost: prikrivanje promjene lokacije sredstva na način da ta promjena ne utječe na način pristupa sredstvu
- Relokacijska transparentnost: prikrivanje premještanja / kretanja sredstva
- Web poslužitelj zadovoljava migracijsku transparentnost. Mijenjanjem lokacije i/ili mrežnog sučelja mijenja se mrežna adresa, ali se ne mijenja simbolička adresa. Web poslužitelj ne zadovoljava relokacijsku transparentnost jer je poslužitelj stacionaran i ne kreće se tijekom pružanja usluge.

1.3. Objasnite pojam skalabilnosti raspodijeljenog sustava.

- Raspodijeljeni sustav je skalabilan ukoliko posjeduje sposobnost prilagodbe povećanom broju korisnika i sredstava, njihovoj rasprostranjenosti te načinu upravljanja sustavom.

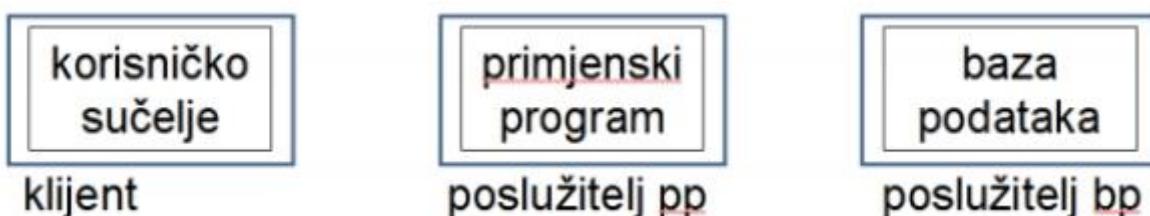
1.4. Kojim aspektima transparentnosti pridonosi sustav imenovanja domena (DNS)?

- Migracijska transparentnost, konkurencijska transparentnost i lokacijska transparentnost.

1.5. Napravite analizu transparentnosti raspodijeljenog sustava elektroničke pošte.

- Primjer gmaila:
Zadovoljava lokacijsku transparentnost jer korisnik pristupa usluzi preko simboličke adrese.
Zadovoljava konkurencijsku transparentnost jer uslugu koristi puno korisnika, a individualan korisnik ima dojam kao da je on jedini. Transparentnost pristupa, prikrivaju se razlike u pristupu sredstvu korisnicima s različitim OS-ima (npr. Mac, MS Windows itd.).
Transparentnost na kvar i replikacijska transparentnost, kopije podataka se čuvaju na drugim poslužiteljima za slučaj kada se jedan poslužitelj pokvari, da drugi može preuzeti posao bez da korisnik primjeti da je došlo do kvara.

1.6. Prikažite i objasnite primjer troredne arhitekture weba.



- Primjer su aplikacije weba, gdje korisnički program koji se izvodi na klijentskom računalu nikad ne pristupa direktno bazi podataka, već posredno preko aplikacije weba. Klijentski program prikazuje korisničko sučelje i komunicira s aplikacijom weba koja obavlja cijelokupnu logiku usluge i pristupa potrebним podacima.

- 1.7. Kakvi bi se problemi pojavili kad bi se više repliciranih poslužitelja weba priključilo na mrežu izravno, a ne putem zastupnika (proxy)?
- Korisnik ne bi znao kojem poslužitelju pristupa i sa svakim sljedećim pristupanjem ne bi bilo zajamčeno pristupanje sadržaju kojem je prethodno pristupio.

- 1.8. Kakvi su tržišni udjeli web-poslužitelja i web-preglednika?

- Wtf?

- 1.9. Što je vrijeme odziva za poslužitelja weba?

- Vrijeme odziva je vrijeme koje je potrebno da poslužitelj procesira zahtjev i da odgovor stigne do korisnika.

2 Procesi i komunikacija

- 2.1. 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 će 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.

- 2.2. 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.

- 2.3. Objasnite zašto tranzijentna sinkrona komunikacija potencijalno pati od problema vezanih uz skalabilnost.

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

- 2.4. 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.

- 2.5. 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.

- 2.6. 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.

2.7. 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.

3 Sloj raspodijeljenog sustava za komunikaciju među procesima

3.3 Poslužitelj je implementiran pomoću socketa TCP s ograničenjem veličine repa za dolazne zahtjeve na instanci klase ServerSocket (tzv. backlog) na 10. Što se događa s novim zahtjevima ako je prilikom dolaska novog zahtjeva u tom repu već na čekanju 10 zahtjeva?

-Zahtjevi se odbacuju.

3.4. Na koji je način moguće ograničiti broj paralelnih konekcija prema višedretvenom poslužitelju kao što je npr.UpperCaseServer?

- Wtf?

3.5. U tablicama su prikazane metode na klijentskoj i poslužiteljskoj strani socketa TCP. Upišite ispravan redoslijed izvođenja metoda u tablice.

Klijent	Klijent
Socket()	Socket()
Write()	Connect()
Read()	Write()
Close()	Read()
Connect()	Close()

Mnemonička metoda ili šalabahter: SCWRC

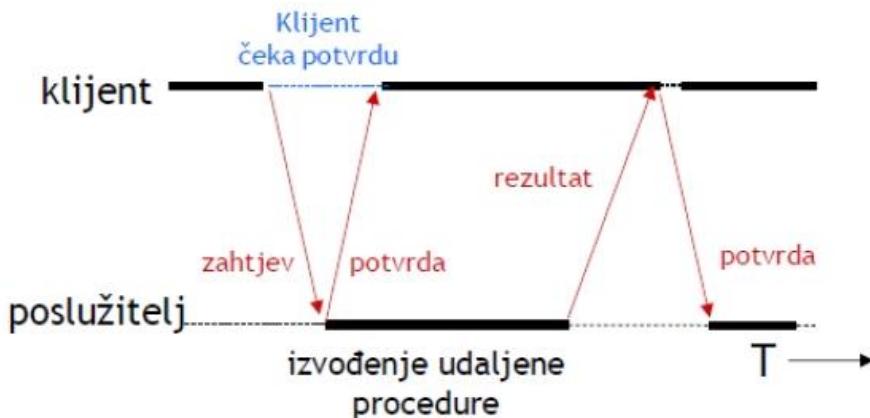
Poslužitelj	Poslužitelj
Listen()	Socket()
Socket()	Bind()
Accept()	Listen()
Write()	Accept()
Read()	Socket()
Close()	Read()
Bind()	Write()
	Close()

Mnemonička metoda ili šalabahter: SBLASRWC

3.6. Može li se pomoću UDP-a implementirati protokol za pouzdanu komunikaciju između klijenta i poslužitelja? Ako može, na koji način?

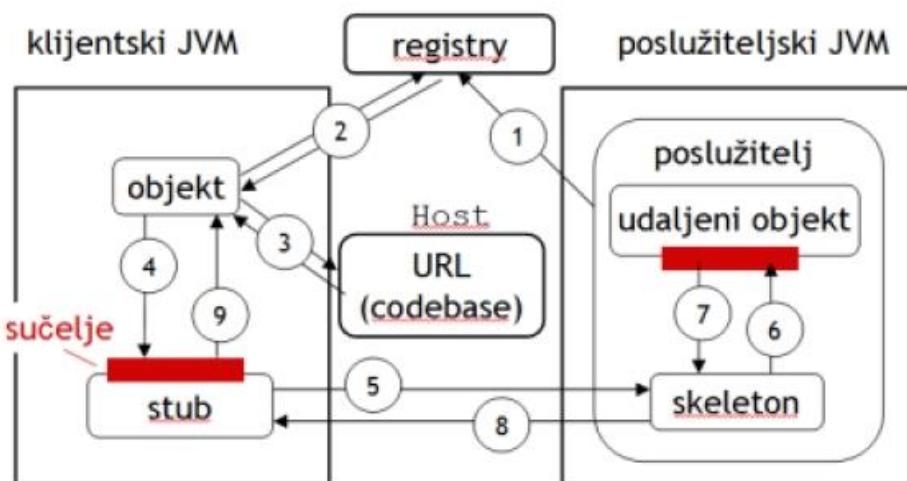
-Može se implementirati pomoću slanja potvrda za svaki UDP paket. Ako korisnik 1 nije dobio potvrdu od korisnika 2 za primitak poslanog UDP paketa nakon određenog vremena, korisnik 1 isponova šalje taj određeni UDP paket.

3.7. Skicirajte tijek komunikacije između klijenta i poslužitelja te objasnite odgođeni sinkroni poziv udaljene procedure RPC (Remote Procedure Call).



- Kod odgođenog sinkronog poziva udaljene procedure, klijent nije blokiran dok čeka rezultat izvođenja, već nastavlja s radom nakon uspješnog primitka potvrde. Kasnije mu poslužitelj šalje rezultat koristeći drugi asinkroni poziv udaljene procedure.

3.8. Skicirajte model pozivanja udaljene metode Java RMI (Remote Method Invocation). Navedite korake u komunikaciji potrebne da bi klijent pozvao metodu dostupnu na poslužitelju, uz pretpostavku da klasa stub već postoji i dostupna je na klijentskoj strani.



- 1) Poslužitelj definira codebase udaljenog objekta i registrira ga pod odabranim imenom.
- 2) Klijent od registra traži referencu na udaljeni objekt, i dobiva je.
- 3) Klijent traži i dobiva klasu stub koristeći codebase.
- 4) Klijent poziva metodu stuba dostupnu na klijentskom računalu.
- 5) Stub serijalizira parametre i šalje ih skeletonu.
- 6) Skeleton deserijalizira parametre i poziva metodu udaljenog objekta.
- 7) Udaljeni objekt vraća rezultat izvođenja metode skeletonu.
- 8) Skeleton serijalizira rezultat i šalje ga stubu.

9) Stub deserijalizira rezultat i dostavlja ga klijentu.

3.9. Ima li smisla implementirati perzistentnu asinkronu komunikaciju pomoću RMI-ja? Zašto?

- wtf?

3.10 Objasnite razliku između transportne adrese i adrese repa u sustavima za komunikaciju porukama?

- Svaki rep ima jedinstveno ime (tzv. adresa repa) neovisno o samoj transportnoj adresi te je potrebna usluga koja povezuje ime repa s transportnom adresom (analogno DNS-u). Adresiranje se izvodi najčešće na nivou sustava, a svaki rep ima jedinstveni identifikator i stoga komunikacija porukama nije anonimna. Pošiljatelj i primatelj ne moraju znati ništa jedan o drugome, no moraju znati adresu repa i format poruke kako bi mogli komunicirati.

3.11 Skicirajte i objasnite primjer komunikacije porukama između dva procesa/objekta (primatelja i pošiljatelja). Kakva je komunikacija porukama s obzirom na vremensku ovisnost primatelja i pošiljatelja?



- U komunikaciji između pošiljatelja i primatelja rep sudjeluje kao posrednik. Pošiljatelju se u načelu garantira isporuka poruke u primateljev rep, ali ne i isporuka poruke primatelju. Primatelj može pročitati poruku iz repa u bilo kojem budućem trenutku.

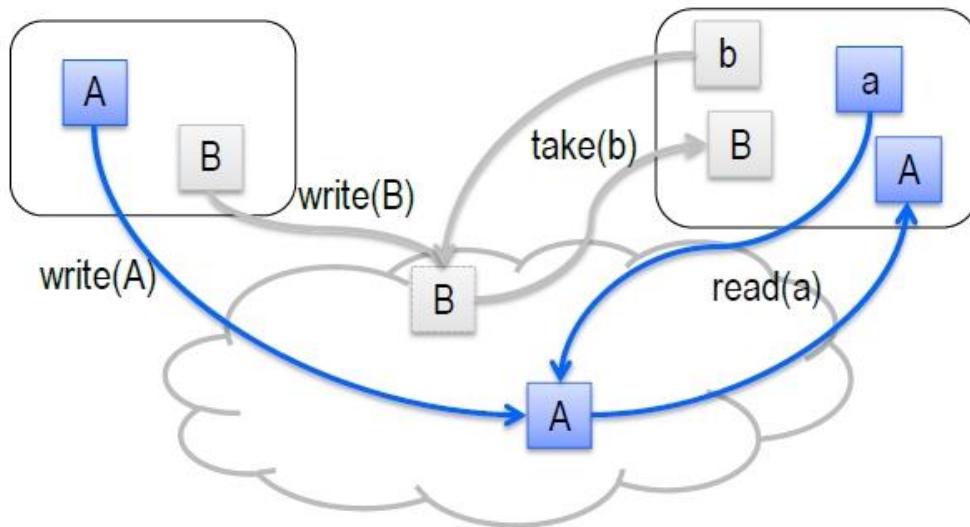
- Primatelj ne mora biti aktivan za vrijeme kada pošiljatelj želi poslati poruku primatelju.

3.12. Navedite i objasnite operacije koje implementira programska infrastruktura dijeljenog podatkovnog prostora.

- write (t): dodaj tuple t u raspodijeljeni podatkovni prostor

- read (s) -> t: vraća tuple t koji odgovara predlošku s

- take (s) -> t: vraća tuple t koji odgovara predlošku s i briše ga iz podatkovnog prostora



3.13. Navedite sličnosti i razlike komunikacije na načelu objavi-preplati i dijeljenog podatkovnog prostora.

- Obje vrste komunikacije su vremenski neovisne zato što pošiljatelj i primatelj ne moraju istovremeno biti dostupni da bi se komunikacija mogla ostvariti. Kod komunikacije porukama pošiljatelj mora znati identifikator odredišta, dok je kod modela objavi-preplati komunikacija anonimna. Komunikacija je perzistentna i asinkrona u oba slučaja. Komunikacija se pokreće na načelu pull kod komunikacije porukama, a na načelu push kod modela objavi-preplati.

3.14. Usporedite preplatu u sustavima objavi-preplati i predložak u sustavima s dijeljenim podatkovnim prostorom. Može li se koristeći navedenu međupremu realizirati tzv. anonimnu komunikaciju (pošiljatelj ne zna adresu primatelja) i zašto?

-Može se kombinirati posrednik iz sustava objavi-preplati s dijeljenim podatkovnim prostorom. Komunikacija u sustavu dijeljenog podatkovnog prostora je anonimna jer se temelji na sadržaju podataka. Dakle, posrednik iz sustava objavi-preplati komunicira s krajnjim korisnicima na temelju dijeljenog podatkovnog prostora.

3.15. Gdje se filtriraju obavijesti u raspodijeljenom sustavu objavi-preplati koji koristi preplavljivanje obavijestima?

-Vrši se u krajnjem posredniku prije isporuke obavijesti lokalnim preplatnicima.

4 Arhitekture web-aplikacija

4.1. Navedite i objasnite najčešće korištene metode protokola HTTP/1.1.

- OPTIONS – služi za informiranje i dohvaćanje mogućnosti resursa i poslužitelja,
- GET – koristi se za dohvaćanje resursa (najčešće u uporabi),
- HEAD – koristi se za dohvaćanje podataka o resursu (npr. veličina, provjera postojanja),
- POST – služi za aktiviranje resursa (npr. slanje podataka obrazaca),
- PUT – postavljanje resursa (npr. promjena podataka u web-uslugama),

- DELETE – brisanje resursa (npr. kod web-usluga),
- TRACE – koristi se za dijagnostiku i
- CONNECT – predviđeno za buduću uporabu.

4.2. Objasnite namjenu jezika CSS. Zašto se pojavila potreba za definiranjem takvog jezika?

- Jezik CSS (*Cascading Style Sheets*) je jezik za definiranje pravila prikaza sadržaja HTML-ovog dokumenta, a nastao je iz potrebe odvajanja sadržaja i prikaza dokumenta.

4.3. Korisnik nakon ispunjavanja obrasca na Web-u odabire opciju *Submit*, čime pošalje podatke Web-poslužitelju na adresu www.tel.fer.hr/obrazac/accept korištenjem protokola HTTP verzije 1.1. Kojim se HTTP zahtjevom šalju podaci poslužitelju i kako je definiran prvi redak zahtjeva?

- Podaci se šalju POST HTTP zahtjevom. Prvi redak je definiran na sljedeći način: POST /obrazac/accept HTTP/1.1

4.4. Objasnite opći format poruka protokola HTTP. Navedite kako glasi potpun i apsolutan URI koji identificira resurs zatražen u zahtjevu, ako prva 2 retka HTTP zahtjeva sadrže sljedeće podatke:

GET /predmet/rassus HTTP/1.1

Host: www.fer.unizg.hr

- Poruka se sastoji od sljedećih elemenata: Početnog retka koji je obvezan, zaglavljiva koja se sastoje od polja (neka polja su obvezna), prazan redak i tijela poruke. Obvezni dijelovi su početni redak, dio zaglavljiva s poljem Host i prazan redak.
- <http://www.fer.unizg.hr/predmet/rassus>

4.5. Objasnite ulogu posredničkog poslužitelja s priručnim spremištem koji se nalazi u mreži između web-preglednika i web-poslužitelja. Gdje se sve može nalaziti u mreži?

- Posrednički poslužitelj s priručnim spremištem služi kao privremeno spremište. S njim se ubrzava posluživanje i manje je se tereti mreža između posredničkog poslužitelja i poslužitelja klijenta. Može se nalaziti u lokalnoj mreži klijenta, javnoj mreži ili u lokalnoj mreži poslužitelja.

4.6. Navedite i objasnite prednosti korištenja posredničkog poslužitelja s priručnim spremištem za korisnika, izvorni poslužitelj i komunikacijsku mrežu.

- Prednosti s korisničke strane su brže učitavanje sadržaja koji je spremljen u priručno spremište i smanjuje se korištenje veze između lokalne mreže i ostatka mreže.
- Prednosti s izvornog poslužitelja je privremeno spremanje izračunatih podataka: posrednički poslužitelj može svježe resurse spremiti u svoje priručno spremište i njih slati kao odgovor te na taj način rasteretiti poslužitelja usluge.
- Prednost za komunikacijsku mrežu je smanjenje korištenih mrežnih resursa.

4.7. Prepostavite da se sjedište weba sastoji od 2 poslužitelja priključena na Internet preko posrednika (*proxy*). Navedite i objasnite svojstva ovog raspodijeljenog sustava.

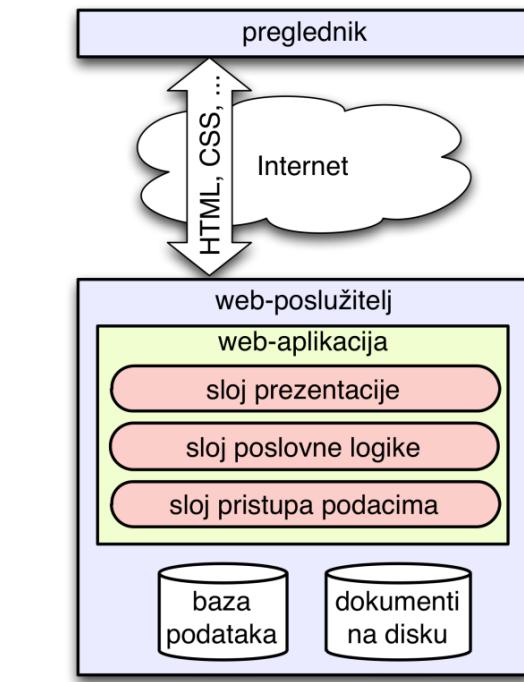
- Zastupnik poslužitelja (*proxy*) posreduje između klijenata i poslužitelja tako da od klijenata prikriva broj i lokaciju poslužitelja te način na koji su povezani (tj. omogućava replikacijsku transparentnost). Također, proxy pruža mogućnost rasterećenja rada poslužitelja tako da nadolazeće zahtjeve preusmjeri na onog poslužitelja koji ja najmanje zaposlen, kriptira ili komprimira podatke i privremeno sprema izračunate podatke i njih šalje kao odgovor, što naravno rasterećuje poslužitelja usluge. Ujedno, može dozvoliti pristup samo nekim uslugama i na taj način poslužitelj usluge ne mora biti toliko siguran jer ga štiti posrednički poslužitelj koji može osiguravati SSL komunikaciju između klijenta i posredničkog poslužitelja, a komunikacija između posredničkog poslužitelja i poslužitelja usluga ne treba biti zaštićena.

4.8. Objasnite razliku između web-aplikacija temeljenih na CGI (Common Gateway Interface) i poslužiteljskim skriptama.

- CGI (Common Gateway Interface) je jednostavno sučelje za pokretanje eksternih programa iz web-poslužitelja na platformski i programski neovisan način. Kod svakog zahtjeva se pokreće novi proces, a podaci između poslužitelja i procesa šalju se preko varijabli okoline i tokova podataka. Nakon svake obrade proces se gasi.
- Poslužiteljske skripte isto dinamički generiraju HTML-dokumente, ali je razlika u tome što se za svaki zahtjev ne pokreće novi proces i na taj način se štede resursi.

4.9. Skicirajte i objasnite slojevitu arhitekturu web-aplikacija.

- Web-aplikacije su obično organizirane u 3 sloja:
 - sloj prezentacije – služi za prikaz informacija (GUI, HTML, klikovi mišem, ...) i za obradu HTTP-zahtjeva;
 - sloj poslovne (domenske) logike – obrađuje podatke koje je dobio od sloja prezentacije. To je glavni dio sustava koji radi ono za što je sustav namijenjen;
 - sloj pristupa podacima – komunicira s bazom podataka i drugim komunikacijskim sustavima te se brine o transakcijama i o pohrani podataka.

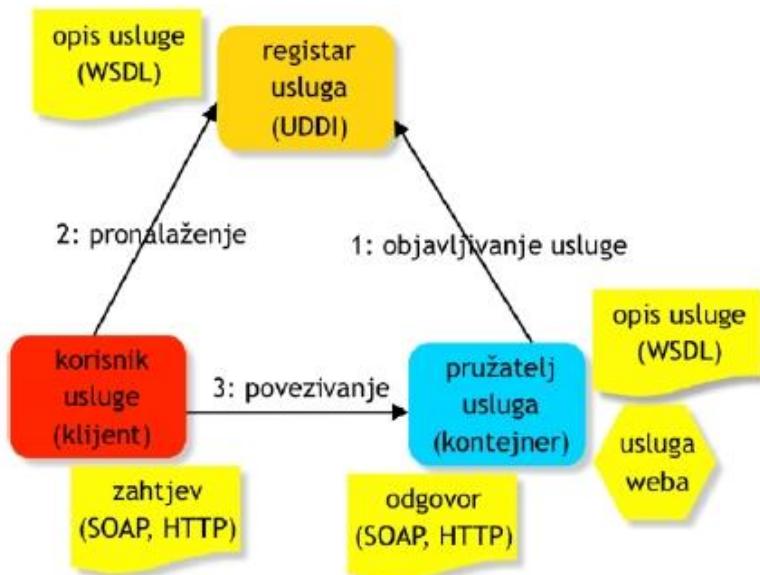


4.10. Koja je prednost korištenja tehnike dugog prozivanja poslužitelja za web-aplikacije?

Prednost je što se u slučajevima kada nema puno poruka ne opterećuje mreža s klijentovim zahtjevima i poslužiteljevim praznim odgovorima. Nego klijent pošalje jedan zahtjev i poslužitelj mu odgovori s porukom tek kada ta poruka stigne od drugog klijenta.

5 Web usluge

5.1. Skicirajte i objasnite arhitekturu web-usluge te interakcije između komponenti web-usluge.



Kada se na pružatelju usluge instalira web-usluga, pružatelj ju prvotno registrira u registru usluga. Koristeći protokol UDDI pružatelj objavi uslugu i registru šalje opis usluge u WSDL-u. Kada klijent želi pronaći uslugu, prvo registru pošalje upit za pronalaženje, a registar mu odgovara opisom usluge koja odgovara zadanim kriterijima. Nakon toga se klijent može povezati s pružateljem usluge jer ima opis usluge. Klijent pomoću protokola HTTP i SOAP šalje zahtjev za pozivom usluge, pružatelj usluge pokreće izvođenje usluge i šalje natrag rezultat u odgovoru HTTP i SOAP protokolima.

5.2. Navedite i ukratko objasnite tri vrste web-usluga.

1. Poziv udaljene procedure – usko povezuje klijenta i poslužitelja jer je usluga usko vezana za programski jezik jer se za svaki programski jezik koristi alat koji automatski iz programskog jezika na svoj način definira elementa WSDL-a.
2. Usluge temeljene na dokumentima / porukama – ne povezuje jako klijenta i poslužitelja odnosno njihove implementacijske jezike jer je fokus na ugovorima koji su propisani WSDL-om.
3. Usluge temeljene na stanju resursa - koristi protokol HTTP pa je sučelje dobro definirano, fokus je na interakciji s resursima koji imaju stanje, a ne na porukama ili operacijama.

5.3. Prikažite arhitekturu i objasnite korištenje usluge Weba.

Valjda isto kao 5.1.?

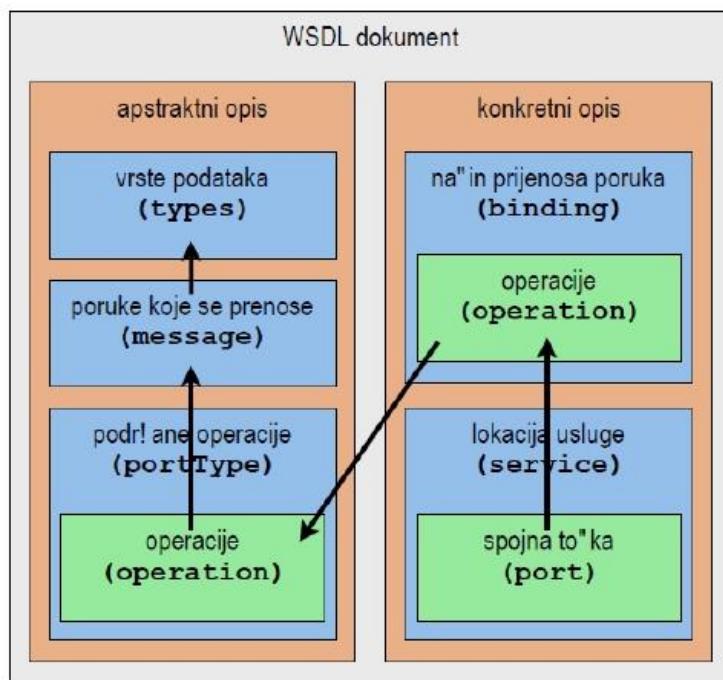
5.4. Navedite dva osnovna načina rada protokola SOAP i objasnite kako se poruka SOAP šalje pomoću protokola HTTP.

Dva osnovna načina rada protokola SOAP su:

- 1) Poziv udaljene procedure – služi za prijenos serijaliziranih parametara i rezultata. Korist ovog načina rada je dobro definirano sučelje i tipovi podataka, te prilagodni kod može biti generiran automatski.
- 2) Razmjena dokumenata / poruka – koriste se XML dokumenti za razmjenu poruka.

5.5. Navedite i objasnite primjenu dokumenta u WSDL-u. Od kojih se dijelova sastoji takav dokument?

Dокумент u WSDL-u se koristi za opis web-usluge. Dokument se sastoji od apstraktnog i konkretnog dijela.



Types – definira vrste podataka neovisne o platformi i jeziku

- 2) Message – definiraju ulazne i izlazne poruke koje se mogu koristiti kao parametri usluga
- 3) Operation – predstavlja jednu operaciju/metodu/proceduru koja je definirana u usluzi, a sastoji se od definicija ulaznih, izlaznih i iznimnih poruka koje se mogu razmjenjivati
- 4) Binding – definira kako je konkretna implementacija povezana s operacijama u apstraktnom opisu i definira formt u kojem de se poruke prenosi
- 5) Service – definirani URI preko kojeg se usluga može pozvati

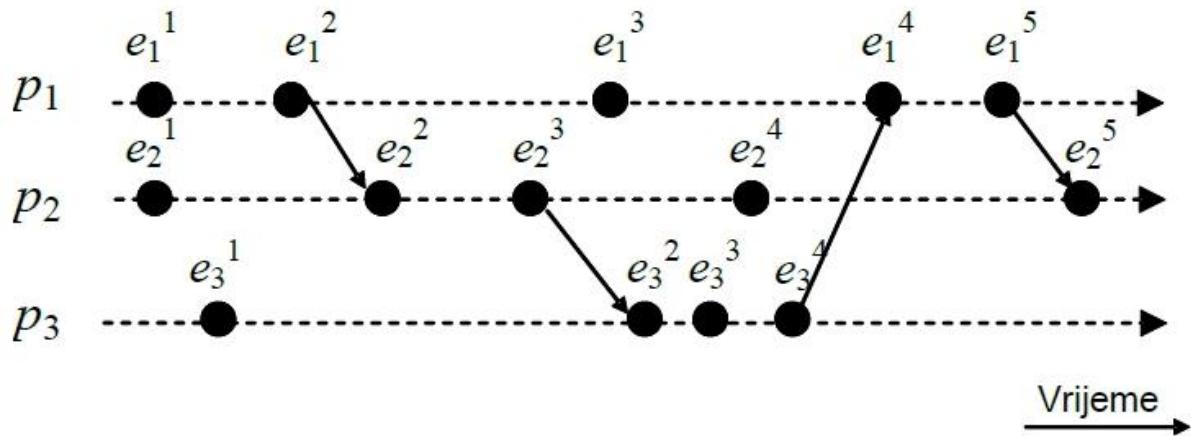
5.6. Objasnite svojstvo idempotentnosti za web-usluge temeljene na REST-u.

Metoda koja ima svojstvo idempotentnosti se može pozvati više puta, ali rezultat će uvijek biti isti.

6. Model raspodijeljenog sustava

- 6.1. Za koje je svojstvo raspodijeljenih sustava značajna komunikacijska složenost algoritama? Zašto?
- Za skalabilnost sustava. Na temelju komunikacijske složenosti možemo zaključiti kako raste generirani promet raspodijeljenog sustava s rastom tog sustava.

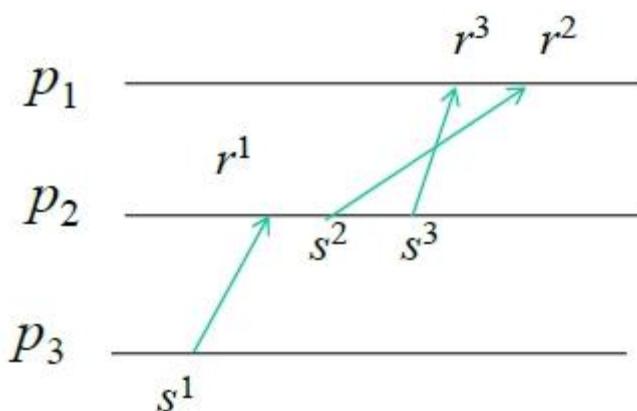
- 6.2. Na temelju primjera procesa sa slike objasnite jesu li sljedeći parovi događaja uzročno povezani ili nisu? a) e_1^3 i e_2^2 i b) e_2^2 i e_1^5 .



- $e_1^3 \rightarrow e_2^2$, zato što:
 $e_1^2 \rightarrow e_2^2$, e_1^2 se dogodio prije e_2^2 , ali e_1^3 se dogodio nakon e_1^2 stoga ne postoji tranzitivna ovisnost i ne e_2^2 i e_1^3 ne razmjenjuju poruke.
- $e_2^2 \rightarrow e_1^5$?
 $e_2^2 \rightarrow e_2^3, e_2^3 \rightarrow e_3^2, e_3^2 \rightarrow e_3^3, e_3^3 \rightarrow e_3^4, e_3^4 \rightarrow e_1^4, e_1^4 \rightarrow e_1^5$. Preko tranzitivnosti $e_2^2 \rightarrow e_1^5$

6.3. Objasnite model komunikacijskog kanala koji se temelji na uzročnoj slijednosti.
 Vrijedi li za sljedeći primjer CO ili non-CO i zašto?

- Kanal koji osigurava uzročnu slijednost (causal ordering, CO) osigurava da uzročno povezani događaji slanja dviju poruka istom primatelju rezultiraju primanjem u slijedu kojim su poslati.
-

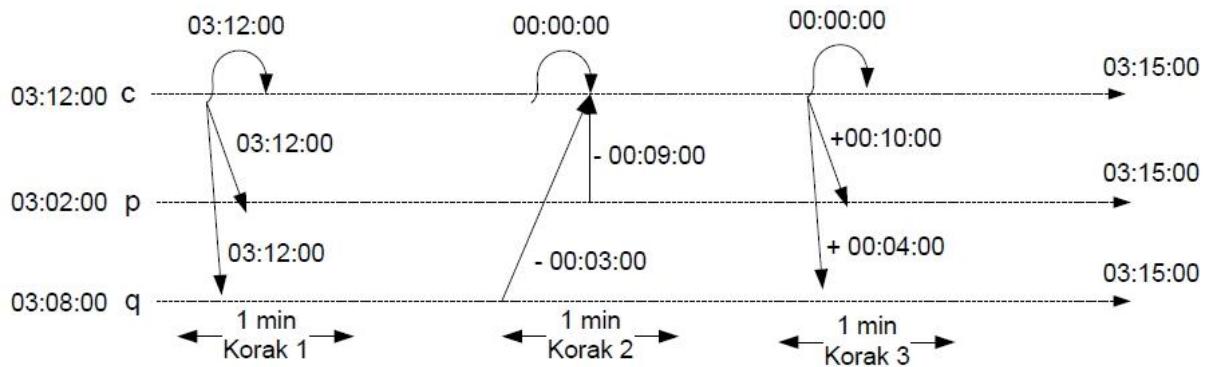


- Non-CO. Proces 2 šalje poruke procesu 1 slijedom s_2, s_3 . Proces 1 umjesto da te poruke dobije istim slijedom kojim su poslane, on ih dobije obrnutim redoslijedom.

7 Sinkronizacija procesa u vremenu

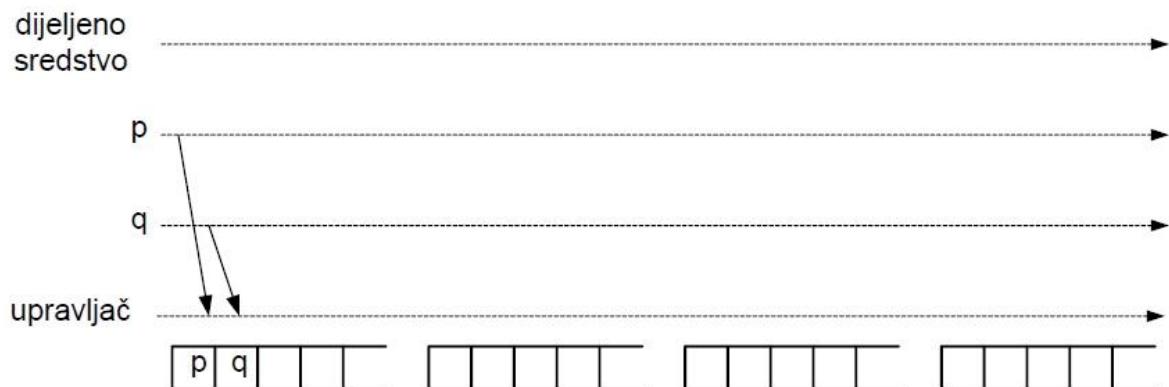
7.1. Prikažite i objasnite korake algoritma Berkeley za usklađivanje satnih mehanizama tri računala u raspodijeljenoj okolini. Računala imaju sljedeće vrijednosti satova $T(p)=03:02:00$, $T(q)=03:08:00$ i

$T(c)=03:12:00$. Upravitelj je treće računalo. Pretpostavite da prijenos poruke između 2 računala traje 1 minuta i da upravitelj koristi svoje lokalno vrijeme kao zajedničko pri usklađivanju satnih mehanizama.

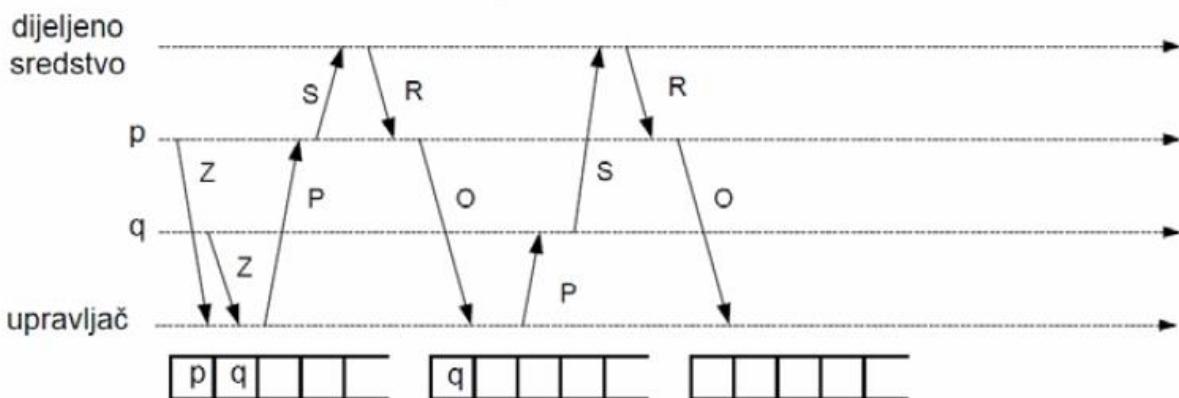


7.2. Opišite postupak međusobnog isključivanja dvaju procesa (p i q) primjenom središnjeg upravljača s repom čekanja tako da nacrtate redoslijed operacija i objasnite ih. Nakon zauzimanja dijeljenog spremnika, proces provodi jednu operaciju čitanja ili pisanja nad dijeljenim spremnikom.

R –Dohvati, S –Spremi, Z –Zauzmi, P –Potvrda, O –Oslobodi



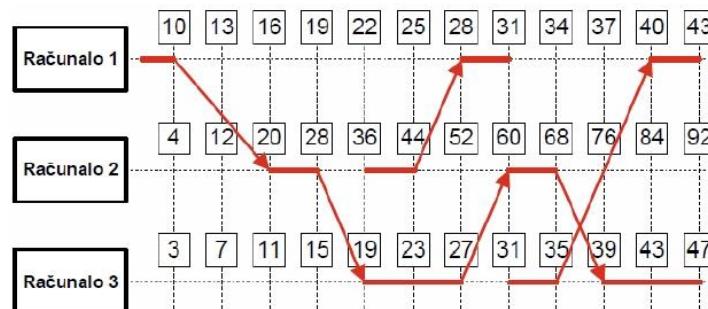
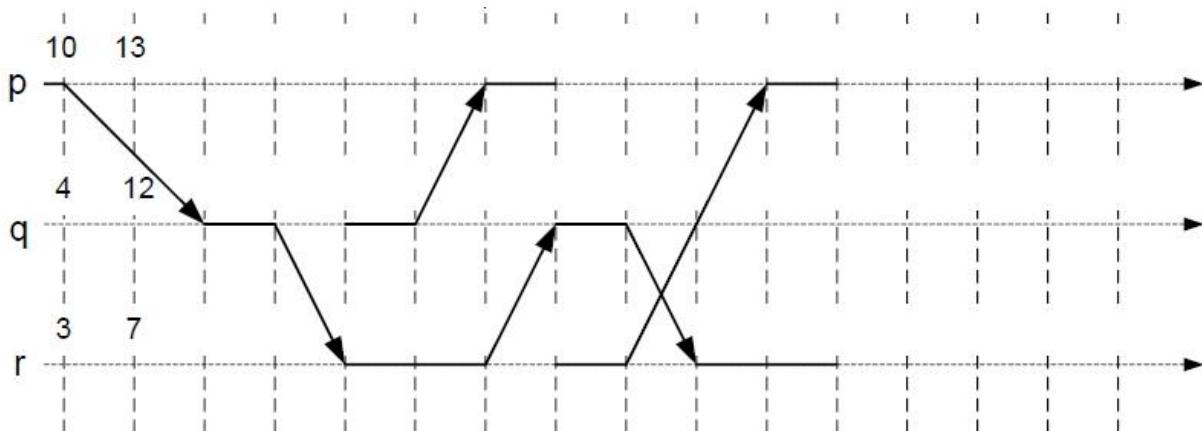
R –Dohvati, S –Spremi, Z –Zauzmi, P –Potvrda, O –Oslobodi



- Proces p šalje zahtjev za zauzimanje sredstva, zahtjev se sprema u rep
- Proces q šalje zahtjev za zauzimanje sredstva, zahtjev se stavlja u rep
- Kako je zahtjev od R0 stigao prije, upravljač šalje potvrdu R0 i uklanja njegov zahtjev iz repa
- Proces p provodi operaciju pisanja

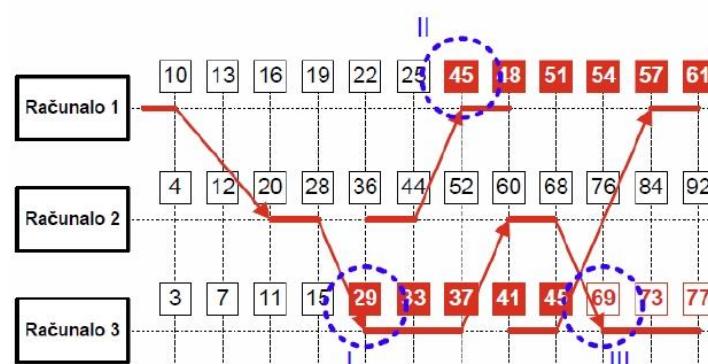
- Proces p prima potvrdu
- Proces p šalje poruku Upravljaču i otpušta pristup
- Upravljač šalje poruku dojave procesu q te mu dodjeljuje pristup dijeljenom spremniku. Iz repa zahtjeva uklanja se zahtjev od procesa p
- Proces q provodi operaciju pisanja
- Proces q prima potvrdu
- Proces q šalje poruku Upravljaču i otpušta pristup

7.3. Za slijed razmjene poruka između tri računala prikazan na slici uspostavite globalni tijek vremena primjenom skalarnih oznaka logičkog vremena. Navedite i opišite trenutke u kojima se ostvaruje korekcija lokalnih satnih mehanizama.



Trenutak I: Računalo 3 prima poruku od računala 2 s oznakom vremena $T_p=28$ koja je veća od lokalne oznake vremena $T_L=19$. Lokalni sat se pomiče na vrijednost $T_p+1=29$.

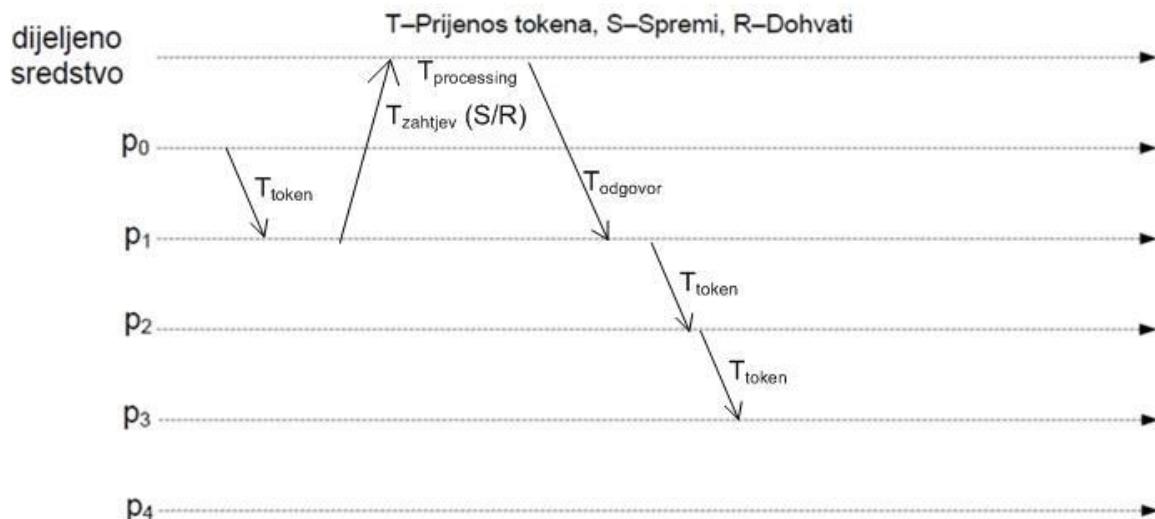
Trenutak II: Računalo 1 prima poruku od računala 2 s oznakom vremena $T_p=44$ koja je veća od lokalne oznake vremena $T_L=28$. Lokalni sat se pomiče na vrijednost $T_p+1=45$.



Trenutak III: Računalo 3 prima poruku od računala 2 s oznakom vremena $T_p=68$ koja je veća od lokalne oznake vremena $T_L=49$. Lokalni sat se pomiče na vrijednost $T_p+1=69$.

7.4. Pet procesa postavljenih na različita računala u raspodijeljenoj okolini ostvaruje međusobno isključivanje primjenom prstena. Vrijeme prijenosa poruke zahtjeva i odgovora pri pristupu dijeljenom sredstvu jednako je 3 ms, vrijeme obrade poruke zahtjeva na sredstvu je 5 ms, vrijeme prijenosa tokena između dva susjedna procesa u prstenu je 2 ms. Kada primi token,

proces može maksimalno jednom ostvariti pristup dijeljenom sredstvu prije nego što proslijedi token idućem susjedu. Naznačite navedena vremena na dijagramu. Koje je minimalno, a koje maksimalno vrijeme čekanja bilo kojeg procesa u prstenu za pristup dijeljenom sredstvu.



Min. vrijeme - U najboljem slučaju, proces koji želi ostvariti pristup čeka $T=0$ sekundi. Naime, taj slučaj nastupa kada proces uđe u stanje u kojem želi ostvariti pristup sredstvu netom prije nego što je primio token.

Max. vrijeme - U najgorem slučaju, proces ulazi u stanje u kojem želi ostvariti pristup sredstvu netom nakon što je proslijedio token svojem susjedu. U tom slučaju, proces mora čekati da svi ostali procesi prime token i ostvare pristup dijeljenom sredstvu. Maksimalno vrijeme čekanja u tom slučaju iznosi $T = 5 * T_T + 4 * (T_z + T_o + T_p) = 10 + 44 = 54$ ms.

8 Konzistentnost i replikacija podataka

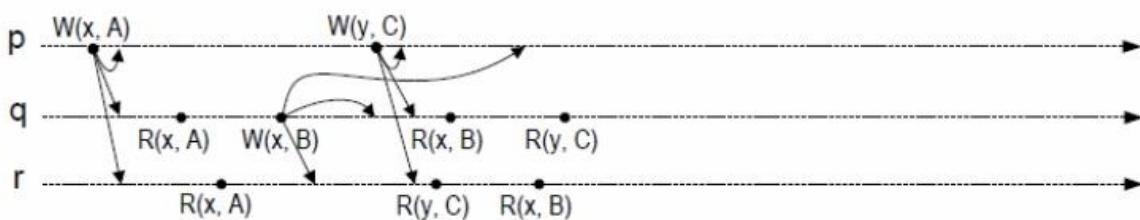
8.1. Objasnite što je replika podatka, a što je nekonzistentnost replike podatka.

Replika podatka je jedna kopija podatkovnog objekta u raspodijeljenoj okolini. Nekonzistentnost replika podataka se javlja kada dvije ili više replika u raspodijeljenoj okolini u nekom trenutku u vremenu se nalaze u različitim stanjima.

8.2. Objasnite što je povezana konzistentnost operacija u raspodijeljenim sustavima? Na primjeru procesa p, q i r prikažite slijed operacija čitanja i pisanja koji je a) u skladu i b) nije u skladu s načelima povezane konzistentnosti.

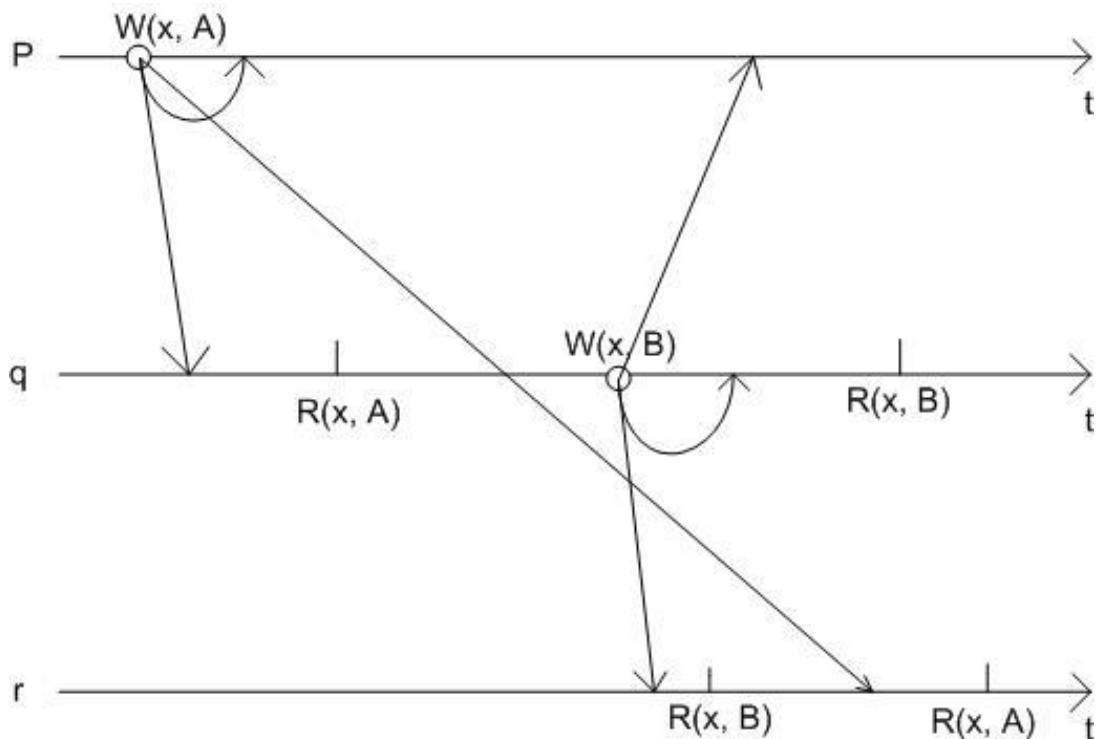
Model povezane konzistentnosti zahtjeva da povezane operacije svi procesi moraju vidjeti u istom redoslijedu. Redoslijed izvođenja povezanih operacija upisivanja vidljiv je svim procesima na jednak način, dok redoslijed izvođenja operacija upisivanja koje nisu povezane, svakom procesu može biti prikazan na drugičiji način.

A)



Proces p zapisuje A na mjesto x, zatim proces q čita A s lokacije x i nakon toga na lokaciju x zapisuje B. Vidi se da su operacije $R(x, A)$ i $W(x, B)$ povezane jer očitana vrijednost A može utjecati na izračun vrijednosti B. Operacije $W(x, B)$ i $W(y, C)$ nisu povezane jer upisuju dva različita podatka na dvije različite lokacije. Dakle, svi procesi moraju vidjeti slijed povezanih operacija u istom redoslijedu tj. mora se prvo izvršiti $R(x, A)$ pa onda $W(x, B)$. Vidimo da se na procesu q tako i izvršava, ali vidimo da je taj redoslijed i na procesu r.

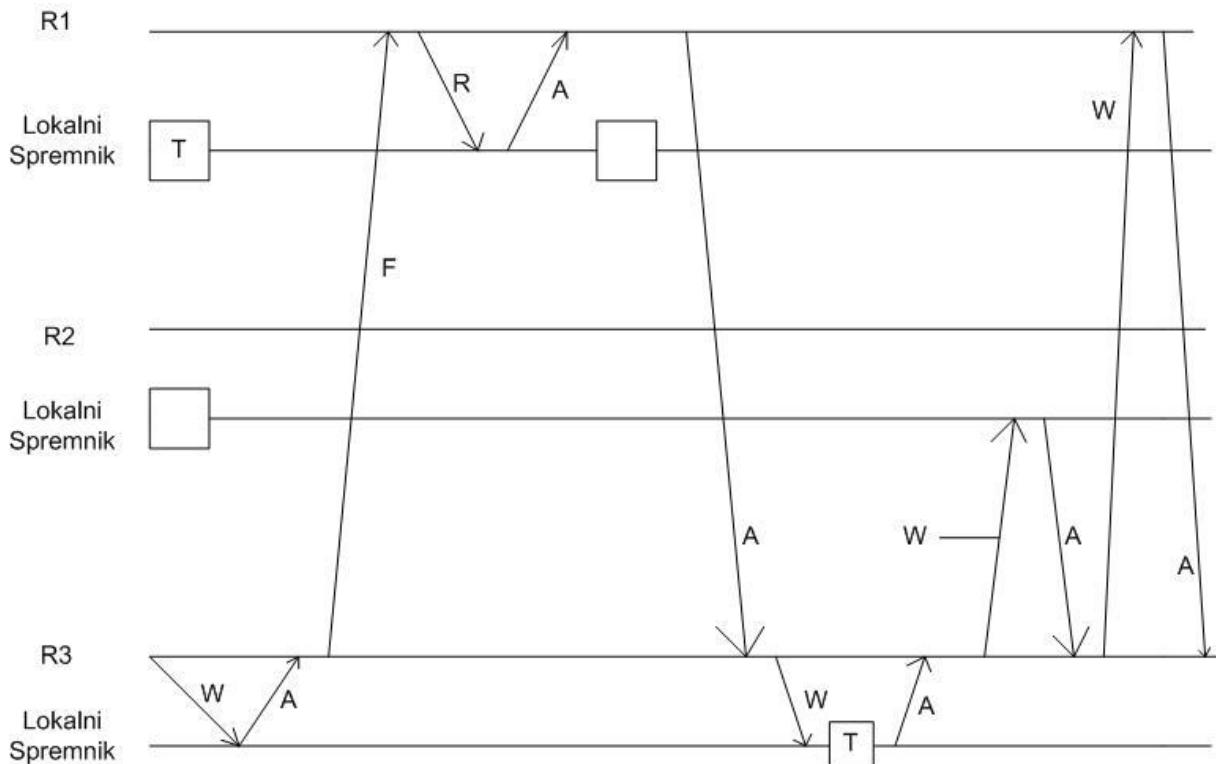
B)



Proces p zapisuje na lokaciju x, podatak A. Zatim proces q čita A s lokacije x i nakon toga zapisuje na lokaciju x podatak B. Vidimo da su operacije $R(x, A)$ i $W(x, B)$ povezane i svaki proces bi trebao vidjeti taj slijed operacija. No, vidimo da proces r prvo zapisuje B na lokaciju x, zatim čita A s lokacije x i zato ovaj primjer ne zadovoljava povezanu konzistentnost.

8.3. **Raspodijeljeni sustav uključuje** tri računala (R_0, R_1, R_2) s lokalnim spremnicima. U lokalnom spremniku računala R_1 nalazi se trajna replika dokumenta, dok se u lokalnom spremniku računala R_2 nalazi obična replika dokumenta. Korisnik putem računala R_3 provodi operaciju pisanja nad dokumentom primjenom postupka lokalnog obnavljanja stanja replike. Skicirajte i objasnite korake postupka.





Korisnik na računalu R3 želi pristupiti promijeniti saržaj podatka (replike). Prvo pristupa svom lokalnom spremniku da provjeri je li se tamo nalazi podatak. Nakon što utvrdi da se ne nalazi od glavnog poslužitelja zatraži repliku. Glavni poslužitelj (R1) dohvati repliku iz svog lokalnog spremnika i proslijedi ju korisniku. Korisnik lokalno obavlja izmjenu replike i on sam postaje glavni poslužitelj. Nakon izmjene obavještava ostale poslužitelje s novom verzijom replike.

8.4. U sustavu replika koji se sastoji od glavnog poslužitelja i n=4 podjednako opterećena pomoćna poslužitelja, odredite metodu održavanja konzistentnosti replika za koju će prosječno mrežno (prometno) opterećenje poslužitelja L biti najmanje. Pri tome prepostavite da korisnike isključivo poslužuju pomoći poslužitelji, da je prosječna frekvencija upita $f_u = 5$ upita/s, prosječna frekvencija promjena $f_p = 1$ promjena/min te da su prosječne veličine upita/odgovora, operacija za promjenu sadržaja i replika $I_p = 1\text{ kB}$, $I_o = 50 \text{ kB}$ i $I_r = 100 \text{ kB}$. Usporedite dobivena opterećenja s centraliziranim slučajem kada korisnike poslužuje glavni poslužitelj.

PUSH metoda s proslijeđivanjem obavijesti o promjenama: Nakon svake promjene, glavni poslužitelj šalje svakom pomoćnom poslužitelju obavijest o promjenama. Za prvi sljedeći korisnički zahtjev, svaki od pomoćnih poslužitelja šalje glavnom poslužitelju upit za novom replikom na koji mu glavni poslužitelj odgovara novom verzijom replike.

$$L_1 = n \cdot f_p \cdot I_p + n \cdot f_p \cdot I_p + n \cdot f_p \cdot I_r = n \cdot f_p \cdot (2 \cdot I_p + I_r) = 6,8 \text{ kB/s}$$

PUSH metoda s proslijeđivanjem operacija za promjenu sadržaja: Nakon svake promjene, glavni poslužitelj šalje svakom pomoćnom poslužitelju obavijest o operacijama koje je potrebno izvršiti nad prethodnom verzijom replike da bi se ona dovela u konzistentno stanje.

$$L_2 = n \cdot f_p \cdot I_o = 3,33 \text{ kB/s}$$

PUSH metoda s proslijedivanjem cjelokupnog sadržaja replika: Nakon svake promjene, glavni poslužitelj šalje novu verziju replike svakom pomoćnom poslužitelju.

$$L_3 = n \cdot f_p \cdot l_r = 6,67 \text{ kB/s}$$

PULL metoda: Nakon svakog upita, pomoćni poslužitelj provjerava kod glavnog poslužitelja je li došlo do promjene stanja replike koju pohranjuje lokalno. U $1/(n \cdot f_p)$ od $1/f_u$ slučajeva će poslužitelj odgovoriti novom verzijom replike, a u $1/(f_u - n \cdot f_p)$ od $1/f_u$ slučajeva će odgovoriti porukom da nije došlo do promjene.

$$L_4 = f_u \cdot l_p + n \cdot f_p \cdot l_r + (f_u - n \cdot f_p) \cdot l_p = 16,6 \text{ kB/s}$$

Centralizirani slučaj: Glavni poslužitelj odgovara replikom na svaki korisnički upit.

$$L_5 = f_u \cdot (l_p + l_r) = 505 \text{ kB/s}$$

$$\text{Min } \{L_1, L_2, L_3, L_4, L_5\} = L_2.$$

9 Otpornost na neispravnosti

9.1. Objasnite razliku između ispada sustava i neispravnosti u sustavu.

-Ispad sustava je stanje sustava koje se detektira kroz nemogudnost korištenja jedne ili više njegovih usluga. Posljedica je neispravnosti te ukazuje na nju. Neispravnost je nedostatak u programskom kodu, oblikovanju sustava ili komunikacijskom kanalu koji može uzrokovati ispad sustava.

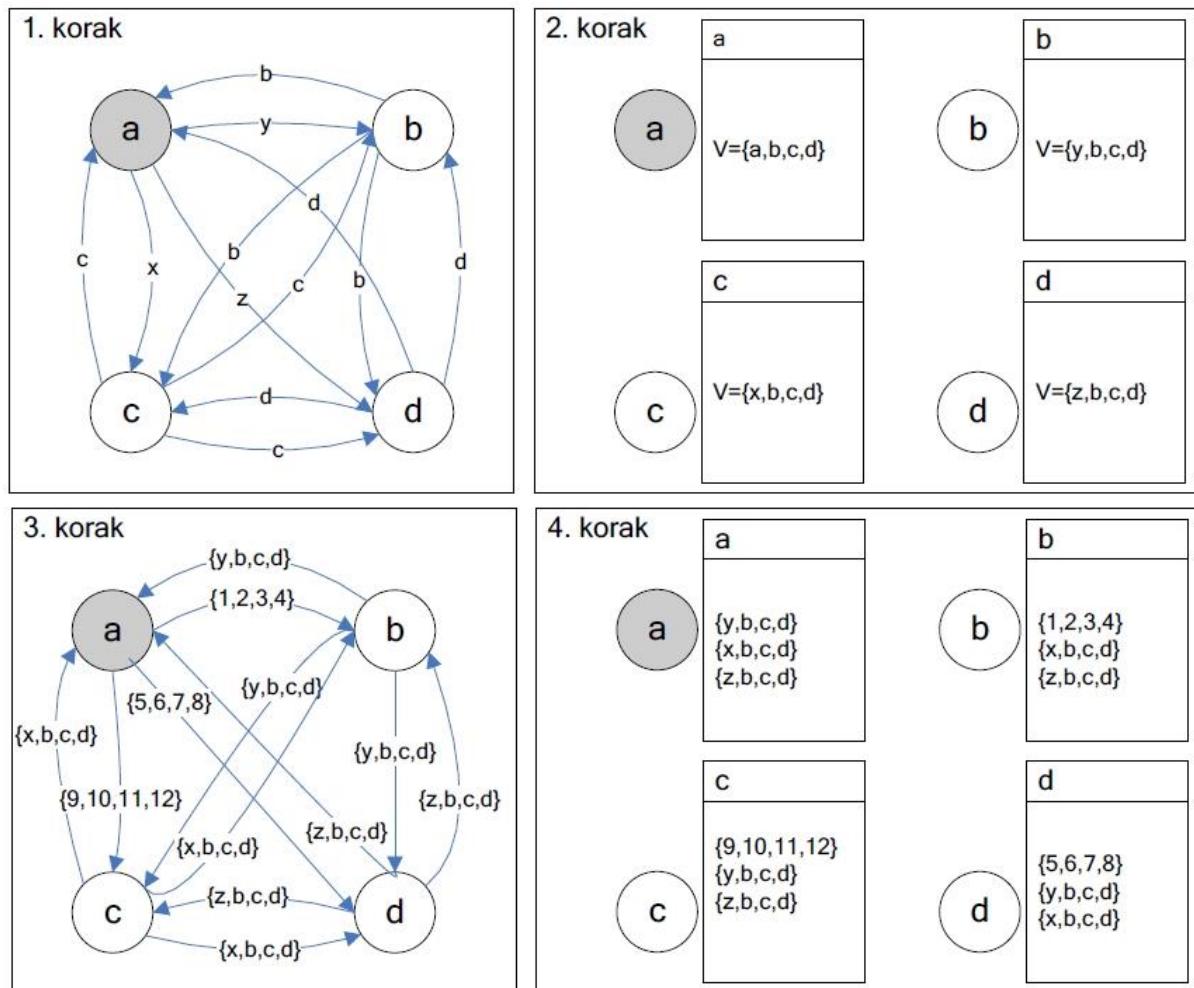
9.2. Prepostavite da grupa procesa treba postići sporazum. U slučaju da su dva procesa grupe u stanju bizantskog ispada, koji je minimalni ukupni broj procesa u grupi za postizanje sporazuma?

$$-N = 3k+1 \text{ gdje je } k \text{ broj procesa u bizantskom ispadu. } N = 3 \cdot 2 + 1 = 7$$

9.3. Objasnite razliku protokola three-phase commit u odnosu na two-phase commit.

-*Three-phase commit* protokol ima jedno stanje više od *two-phase commit* protokola. To stanje je PRECOMMIT. U *two-phase commit* protokolu postoji problem blokirajućeg stanja. Stanje PRECOMMIT riješava taj problem. Koordinator nakon odluke za izvođenje operacije šalje poruku PREPARE_COMMIT na koju procesi odgovaraju s READY_COMMIT. Nakon što primi poruku READY_COMMIT od svih procesa, koordinator šalje GLOBAL_COMMIT. Ako koordinator ispadne, procesi se međusobno mogu dogovoriti koja im je sljedeća akcija i time je riješen problem blokirajućeg stanja.

9.4. U grupi od 4 procesa (p_1, p_2, p_3 i p_4) proces p_1 je neispravan (prepostavite bizantski ispad). Grupa procesa želi postići sporazum o identifikatorima ostalih procesa grupe. U koracima 1 i 3 procesi međusobno razmjenjuju podatke, a u koracima 2 i 4 prikupljaju i analiziraju primljene podatke. Nacrtajte na slici podatke koje procesi razmjenjuju u koracima 1 i 3, a za korake 2 i 4 navedite podatke koje pojedini proces ima na raspolaganju radi donošenja odluke o sporazumu.



10 Vrednovanje performansi raspodijeljenih sustava

10.1 Disk za trajno spremanje podataka ispunjava 50 zahtjeva u sekundi. Srednje vrijeme obrade zahtjeva operacija pisanja i čitanja je 10 ms. Disk ima prosječno 1 zahtjev u repu. Koliko je prosječno vrijeme čekanja na obradu zahtjeva?

Propusnost sustava je $X = 50 \text{ z/s}$.

Srednje vrijeme obrade zahtjeva je $S = 10 \text{ ms/z}$.

Broj zahtjeva u repu je $Q = 1 \text{ z}$.

Vrijeme zadržavanja zahtjeva u sustavu je $R = Q / X = (1 \text{ z}) / (50 \text{ z/s}) = 20 \text{ ms}$.

Vrijeme zadržavanja R uključuje vrijeme čekanja u repu (W) i vrijeme obrade zahtjeva (S): $R = W + S$.

Vrijeme čekanja na obradu je $W = R - S = 20 \text{ ms} - 10 \text{ ms} = 10 \text{ ms}$.

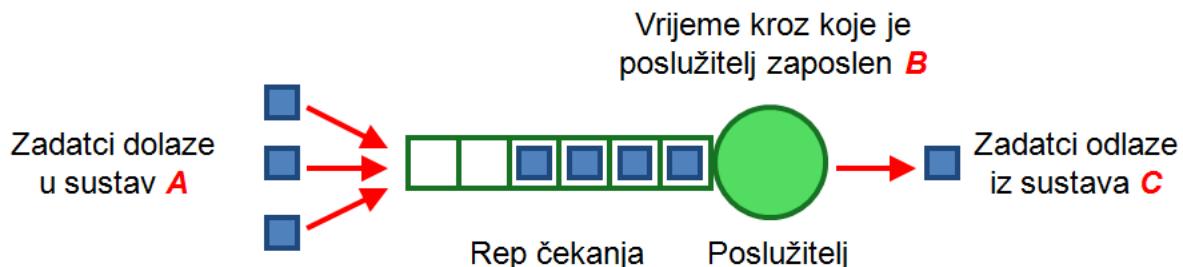
10.2 Web aplikacija uključuje podršku korisnicima putem chat usluge. Kupci sami odabiru jedan od 10 repova čekanja. Mjerenja pokazuju da zahtjevi prosječno dolaze 3 upita u minuti te da svaki kupac prosječno čeka 3 minute u repu i prosječno provodi 2 minute u konverzaciji. Koliko je srednje vrijeme zadržavanja kupaca za zadani sustav?

$$T_w = 3 \text{ min}$$

$$T_s = 2 \text{ min}$$

$$T = T_s + T_w = 5 \text{ min}$$

10.3 Prikažite elemente osnovnog modela repa čekanja. Koje su osnovne veličine, a koje izvedene u modelu repa čekanja? Kako je definirano stacionarno stanje sustava?

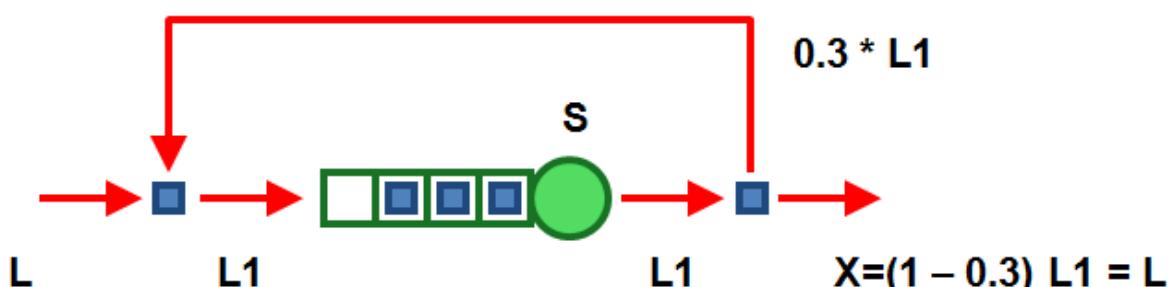


Osnovne veličine modela su vrijeme promatranja (T), broj dolazaka (A), broj odlazaka (C) i vrijeme zaposlenosti poslužitelja (B).

Izvedene veličine modela su ulazni ritam ($L=A/T$), izlazni ritam ($X=C/T$), srednje vrijeme posluživanja ($S=B/C$) i zaposlenost poslužitelja ($U=B/T$).

U stacionarnom stanju sustava je $X = L$.

10.4 Upiti dolaze na poslužitelj s učestalošću od 12 upita u sekundi te zahtijevaju 0,75 sekundi za obradu. Za 30 % paketa dogodi se pogreška pri obradi te se oni moraju ponovno obraditi. Izračunajte koliko vremena paket prosječno provede u sustavu?



$$\text{Broj pristiglih upita u sekundi } L = 0.5 \text{ p/s}$$

$$\text{Prosječno vrijeme obrade upita } S = 0.75 \text{ s/p}$$

$$\text{Vjerovatnost pogreške pri obradi } p = 0.3$$

$$L_1 = p L_1 + L \Rightarrow L_1 = L / (1 - p)$$

$$L_1 = L / (1 - p) = 0.5 / 0.7 = 0.714 \text{ p/s}$$

Prosječna zaposlenost poslužitelja

$$U = L_1 * S = 0.714 \text{ p/s} * 0.75 \text{ s/p} = 0.536 (53.6\%)$$

Srednje vrijeme čekanja u repu

$$W = S * U / (1 - U) = 0.866 \text{ s/p}$$

Srednje vrijeme zadržavanja na poslužitelju

$$R_1 = W + S = 0.866 \text{ s/p} + 0.75 \text{ s/p} = 1.616 \text{ s/p}$$

Prosječno vrijeme zadržavanja u sustavu $R = R_1 / (1 - p) = 2.31$

11 Sustavi s ravnopravnim sudionicima

11.1. Usporedite svojstva centraliziranih i decentraliziranih raspodijeljenih sustava na primjeru.

-Centralizirani raspodijeljeni sustavi temelje se na modelu klijent-poslužitelj. U sustavu postoji koordinator koji prihvata zahteve i raspodjeljuje ih među ostalim procesima u sustavu. Osnovna prednost centraliziranih sustava u smislu performansi sustava je vrlo kratko vrijeme odziva sustava. No, njegova mana je jedinstvena točka ispada sustava (koordinator) i velik broj računala potreban za izvršavanje posla. Primjer takvog sustava je web tražilica. S obzirom da su kolekcije dokumenata prilično velike, potreban je i veliki broj računala za održavanje njihovog indeksa. Npr. za 100 TB tekstualnih dokumenata generira se indeks veličine 25 TB za čije održavanje treba oko 3.000 računala. Stoga je potrebna infrastruktura izrazito složena i skupa, a generira i izrazito visoke troškove održavanja.

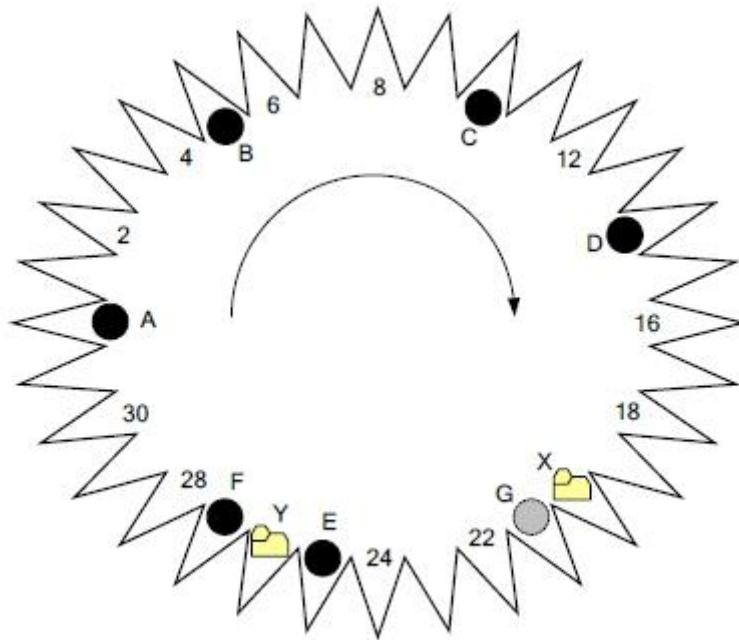
Gnutella je primjer potpuno decentraliziranog sustava koji pretražuje susjedne čvorove s ciljem pronađaska željene datoteke. Sustav se sastoji od peerova koji su međusobno ravnopravni tj. korisnici su ujedno i poslužitelji i klijenti. Svi peerovi sudjeluju u procesu pretraživanja (ne postoji centralizirani indeks), a posebno je pogodno rješenje za pronađenje datoteke koje su replicirane na velikom broju peerova. Ne postoji posebna infrastruktura niti potreba za održavanjem sustava, a time niti jedinstvena točka ispada. Glavni nedostatak ovog rješenja je velika količina generiranog mrežnog prometa pri pretraživanju, a sustav ne može garantirati pronađazak tražene datoteke.

11.2. Kako se izvodi pretraživanje kod strukturiranih, a kako kod nestrukturiranih sustava P2P (peer-to-peer)? Koji od ovih sustava su skalabilni i zašto?

-U nestrukturiranim sustavima P2P, podatak je pohranjen na peeru koji ga kreira, a njegova kopija može biti pohranjena i na nekim drugim peerovima u mreži. Zbog toga se u ovim sustavima pretraživanje izvodi preplavljanjem mreže i slučajnim izborom (*random walk*). Kod strukturiranih sustava P2P, podatak je pohranjen na peeru koji je zadužen za ključ tog podatka. Pretraživanje se provodi traženjem podatka po njegovom ključu. Strukturirani sustavi su skalabilni zato što se kod njih pretraživanje odvija u $\log(n)$ koraka, gdje je n broj peerova u mreži.

11.3. Na slici je prikazana mreža Chord koja se sastoji od 6 čvorova (A, B, C, D, E i F) i koristi prostor identifikatora duljine $N=32$ (dovoljno je $m=5$ bita za kodiranje). Ukoliko je $H_1(A)=0$, $H_1(B)=5$, $H_1(C)=10$, $H_1(D)=14$, $H_1(E)=25$ i $H_1(F)=27$, odgovorite na sljedeća pitanja:

- Definirajte tablice usmjeravanja na čvorovima A i F.
- Na kojem će se čvoru pohraniti podatak X s ključem $H_2(X)=20$?
- Odredite slijed čvorova preko kojih se usmjerava upit od čvora A s ciljem pronađaska podatka Y s ključem $H_2(Y)=26$.
- Dodan je novi čvor G ($H_1(G)=21$) u mrežu. Što će se promijeniti u tablici usmjeravanja čvora A?



a)

Čvor A:

$i = 0, 1, 2, 3, 4$ (i ide od 0 do $m-1$)

$(k+2^i)$

$k = H(A) = 0$

$i = 1$

$(0+1) = 1$, nema čvora na jedan. Prvi sljedeći je 5 tj. čvor B.

$(0+2) = 2$, nema čvora na 2. Prvi sljedeći je 5 tj. čvor B.

$(0+4) = 4$, nema čvora na 4. Prvi sljedeći je 5 tj. čvor B.

$(0+8) = 8$, nema čvora na 8. Prvi sljedeći je 10 tj. čvor C.

$(0+16) = 16$, nema čvora na 16. Prvi sljedeći je 25 tj. čvor E.

Tablica usmjerenja čvora A:

0	5
1	5
2	5
3	10
4	25

Čvor F

$i = 0, 1, 2, 3, 4$ (i ide od 0 do $m-1$)

$(k+2^i)$

$k = H(F) = 27$

$i = 1$

$(27+1) = 28$, nema čvora na 28. Prvi sljedeći je 0 tj. čvor A.

$(27+2) = 29$, nema čvora na 28. Prvi sljedeći je 0 tj. čvor A.

$(27+4) = 31$, nema čvora na 31. Prvi sljedeći je 0 tj. čvor A.

$(27+8) = 35 = (\text{aritmetika modulo } 31) = 32 + 3 = 0+3$, nema čvora na 3. Prvi sljedeći je 5 tj. čvor B.

$(27+16) = 43 = 31 + 11$, nema čvora na 11. Prvi sljedeći je 14 tj. čvor D.

Tablica usmjerenja čvora F:

0	0
1	0
2	0
3	5
4	14

b)

$H_2(X) = 20$

Nema čvora na 20, prvi sljedeći je 25. Podatak X će se pohraniti na čvoru E.

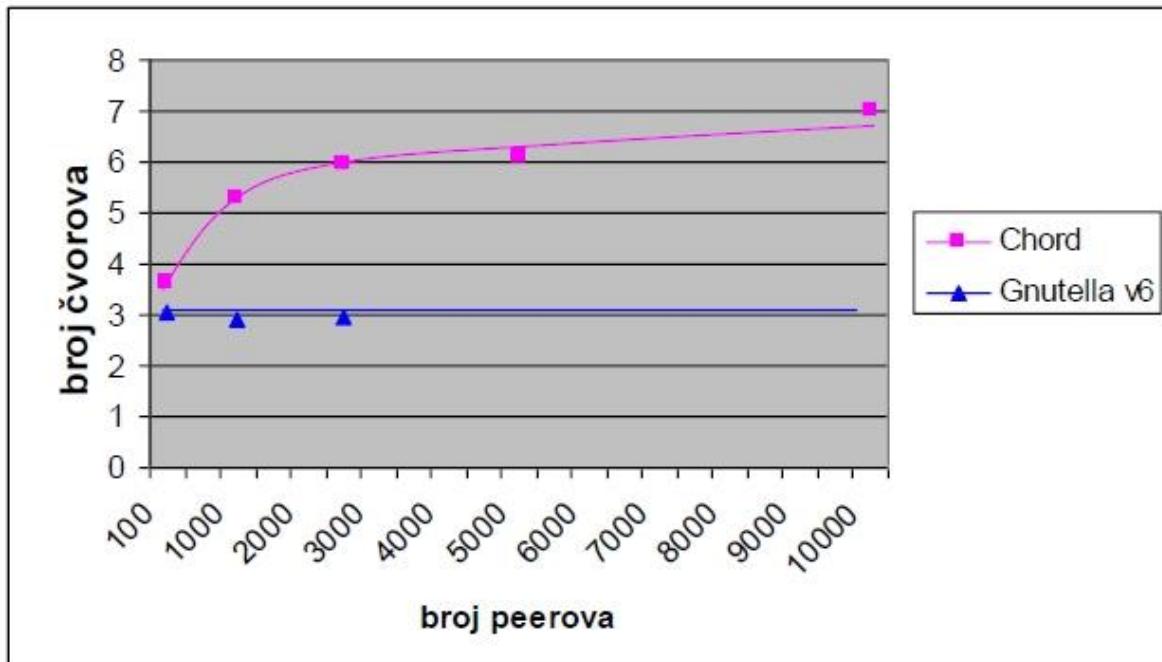
c)

A čvor: $\text{lookup}(26) \rightarrow A$ gleda u tablicu usmjerenja i najveći broj kojeg ima je 25. Proslijedi upit njemu. Čvor 25 proslijedi upit čvoru zaduženom za 26, a to je njegov sljedbenik 27.

d)

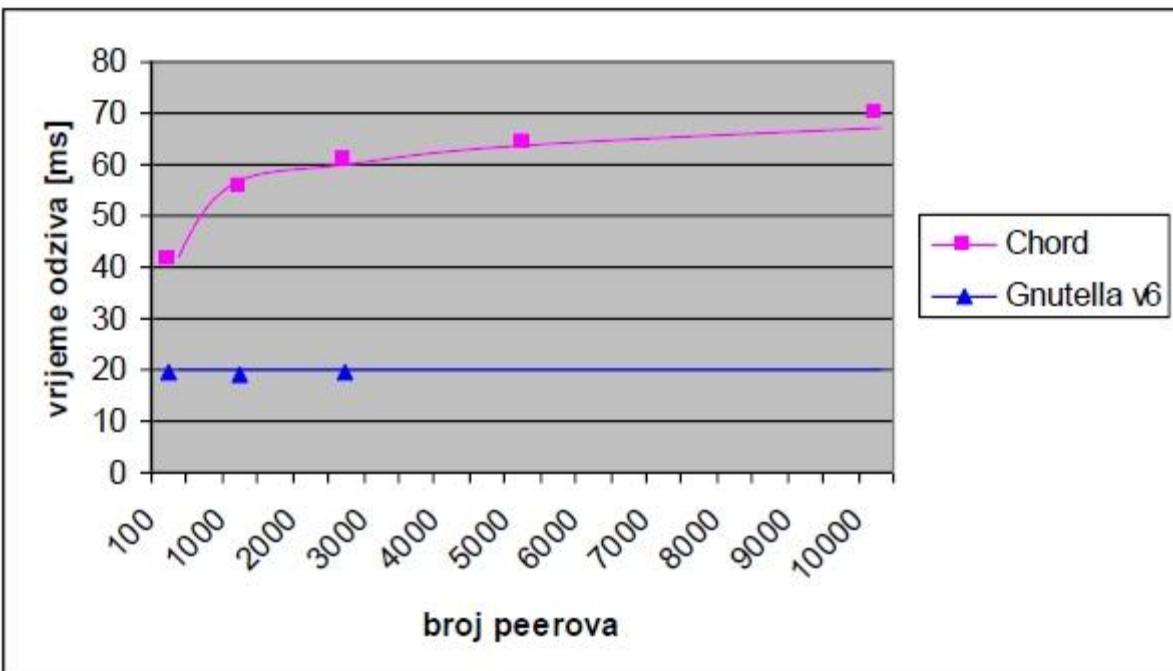
Zadnji redak tablice usmjerenja čvora A će se promijeniti iz 25 u 21.

11.4. Komentirajte rezultate eksperimenta kojim se ispituje svojstvo skalabilnosti protokola Gnutella i Chord u statičnom scenaruju. Kako objašnjavate krivulje kojima se prikazuje prosječan broj čvorova po upitu i prosječno vrijeme odziva?



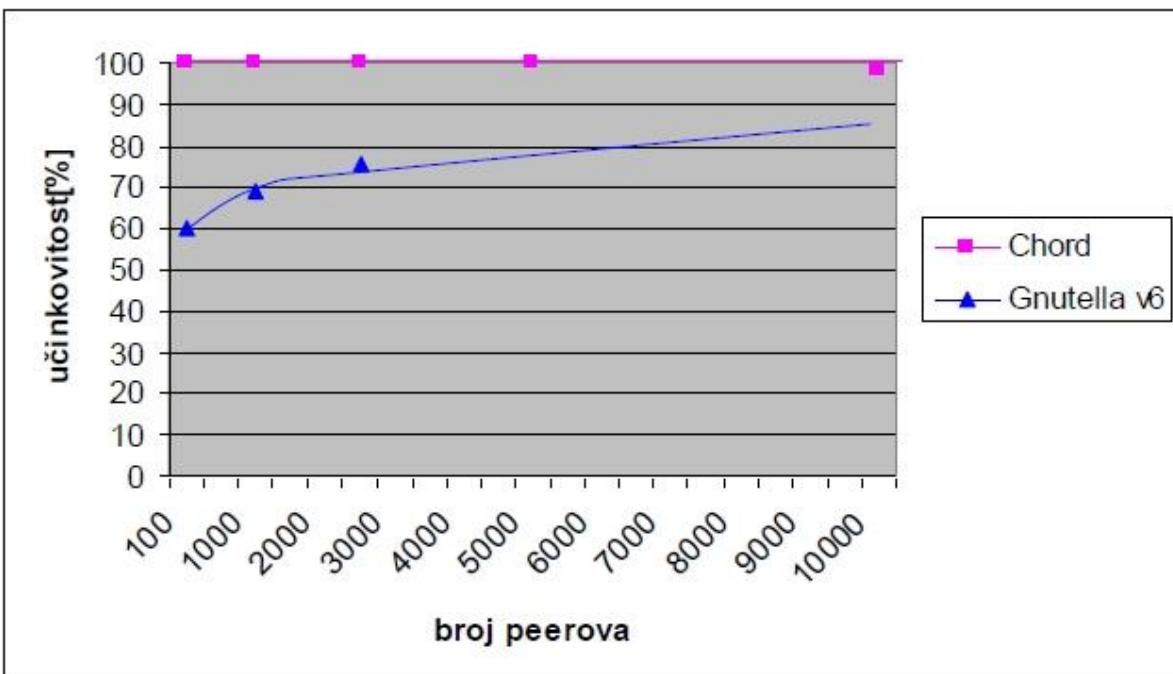
Slika 11.15. Prosječan broj kontaktiranih čvorova po upitu

Vidljivo je da s povećanjem broja peerova u mreži Chord raste broj skokova i vrijeme odziva. Taj porast nije linearan, već se uočava logaritamski porast, što dokazuje da je složenost pretraživanja kod Chorda $O(\log n)$ pri čemu je n broj peerova u mreži. Učinkovitost pretraživanja, tj. postotak uspješno odgovorenih upita je velika što pokazuje slika 11.17.



Slika 11.16. Prosječno vrijeme odziva

Iz slike 11.15. i 11.16. vidljivo je da je kod Gnutelle v6 gotovo jednako vrijeme odziva i broj skokova bez obzira na broj peerova u sustavu, međutim učinkovitost pronaleta podataka raste s povećanjem broja peerova zbog veće povezanosti peerova u mreži što je uočljivo iz slike 11.17.



Slika 11.17. Postotak uspješno odgovorenih upita

12 Grozdovi i spletovi računala

12.1. Skicirajte i ukratko objasnite slojevitu arhitekturu spleta računala.



-**Sloj osnovnih sredstava** upravlja osnovnim funkcionalnostima spleta računala. On implementira lokalne operacije koje su specifične svakom sredstvu po vrsti i implementaciji te omogućuje korištenje tih operacija na višim slojevima.

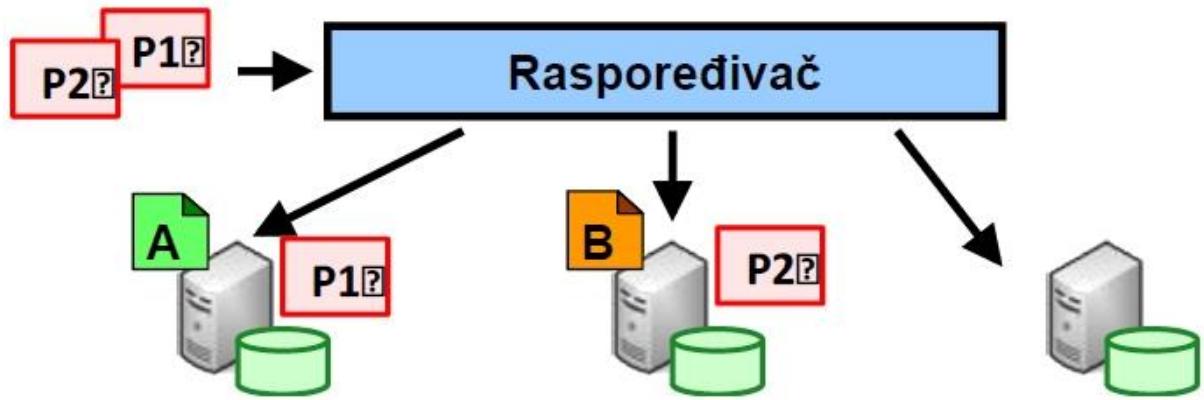
Sloj komunikacijskih protokola definira protokole komunikacije i autentikacije koji su potrebni za transakcije u spletu.

Sloj protokola za sredstva omogućuje korisniku interakciju s udaljenim resursima i uslugama. On definira protokole za sigurno pregovaranje, pokretanje, praćenje, kontrolu, obračun (engl. accounting) i naplatu dijeljenih operacija i individualnih resursa.

Sloj zajedničkih usluga definira protokole i usluge koji su zaduženi za upravljanje grupom sredstava, a ne za pojedino sredstvo.

Na vrhu se nalazi **sloj korisničkih aplikacija**.

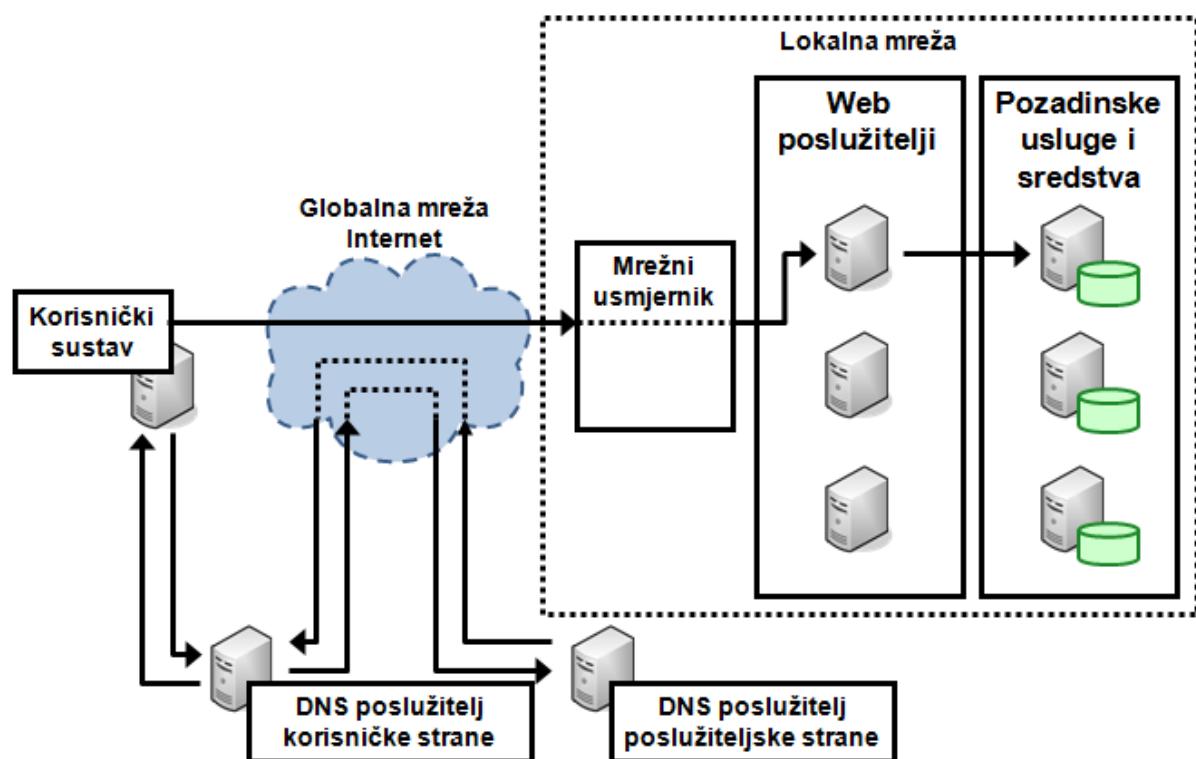
12.2. Na primjeru opišite značajke raspoređivanja zasnovanog na korištenju prostorne lokalnosti.



Slika 12.7. Primjer prostorne lokalnosti

-Kod prostorne lokalnosti, poslovi se raspoređuju na čvorove koji sadrže podatke potrebne za izvođenje posla. Drugim riječima, poslovi se približavaju podacima. Na primjeru sa slike, proces P1 koristi podatke A. Proces P1 se šalje na obradu na čvor na kojem su smješteni podaci A, a proces P2 se šalje na čvor na kojem su smješteni podaci B.

12.3. Prikažite i opišite elemente modela grozda računala.



Korisnički sustav je aplikacija kojom korisnik ostvaruje pristup i koristi sredstva i usluge na grozdu računala.

DNS poslužitelj korisničke strane je poslužitelj pomoću kojeg korisnički sustav razlučuje adrese udaljenih računala na Internetu.

DNS poslužitelj poslužiteljske strane je poslužitelj koji razlučuje adrese poslužitelja u lokalnoj mreži.

Mrežni usmjernik je uređaj koji prihvata, analizira i usmjerava pristigle zahtjeve.

Web poslužitelji i pozadinska sredstva i usluge su osnovni elementi grozda računala.

MI 2014

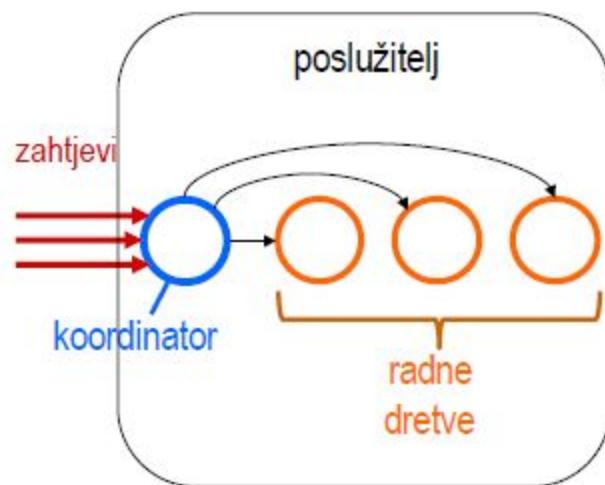
1. Navedi i u jednoj rečenici definiraj 3 obilježja rassusa po želji.

PARALELNE AKTIVNOSTI - autonomne komponente sustava istodobno izvode više aktivnosti

KOMUNIKACIJA PORUKAMA - bez dohvaćanja iz memorije

DIJELJENJE SREDSTAVA - zajedničkim sredstvima pristupa više komponenata sustava

2. Skiciraj model višedretvenog poslužitelja s koordinatorom i radnim dretvama. Objasni tijek komunikacije od trenutka primitka zahtjeva do slanja odgovora.



Koordinator prima zahtjeve, kreira novu dretvu koju pridjeli zahtjevu i novom radnom objektu, a on obrađuje zahtjev i komunicira s klijentom.

3. Može li se pomoću UDP-a implementirati protokol za pouzdanu komunikaciju klijent - poslužitelj? Ako može, objasni kako.

Može, na način da se šalju potvrde za svaki paket. Tada će se paket ponovno poslati, ukoliko pošiljatelj ne dobije potvrdu.

4. Objasni je li implementacija višedretvenog poslužitelja s koordinatorom i radnim dretvama iz prethodnog zadatka skalabilna ili ne. Zašto?

Skalabilnost se omogućuju pomoću višestrukosti (više komponenata) - koordinator + radne dretve. Zbog toga bismo mogli zaključiti da je ova implementacija skalabilna.

5. Odaberi među ponuđenim ona koja upisuju dijeljeni podatkovni prostor.

vremenska ovisnost / vremenska neovisnost

perzistentna / tranzientna

sinkrona / asinkrona

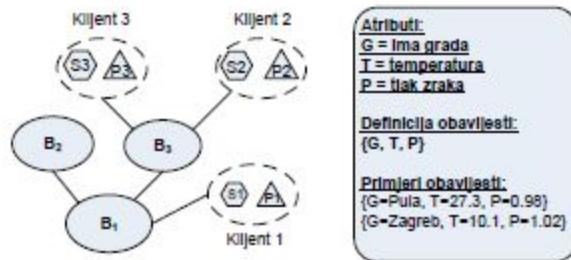
push / pull

6. Objavi-preplati, sličan zadatak kao ovaj dolje iz zadatka za vježbu, ali bio je zadan algoritam preplavljanja preplatama, a ne obavijestima, uz 4 posrednika i 4 klijenta. Pita se kako su u određenim trenucima, nakon generiranja određenih preplat, definirane preplate na posrednicama i kojim posrednicima se proslijeđuje preplata.

Zadatak

3.4

Raspodijeljeni sustav objavi-preplati, u kojem se koristi algoritam preplavljanja obavijestima, sastoji se od 3 posrednika i 3 klijenta kako je prikazano slikom. Svaki klijent u sustavu ima ulogu pretplatnika i objavljavača. Odgovorite na sljedeća pitanja:



- U trenutku t1 klijent 1 generira preplatu $s1=\{G=Zagreb, T<15.5, P>0.98\}$. Napišite oznake svih posrednika na kojima se pohranjuje ova preplata. Preplata se pohranjuje na posredniku B1.
- U trenutku $t2 > t1$ klijent 2 generira preplatu $s2=s1$. Napišite oznake svih posrednika na kojima se pohranjuje ova preplata. Preplata se pohranjuje na posredniku B3.
- U trenutku $t3 > t2$ klijent 3 generira obavijest $p1=\{G=Zagreb, T=-2.2, P=1.01\}$. Objasnite točan redoslijed kojim će se ova obavijest proširiti sustavom i biti isporučena zainteresiranim klijentima.
 $P3 \rightarrow B3 \rightarrow S2 \rightarrow B1 \rightarrow S1 \rightarrow B2$

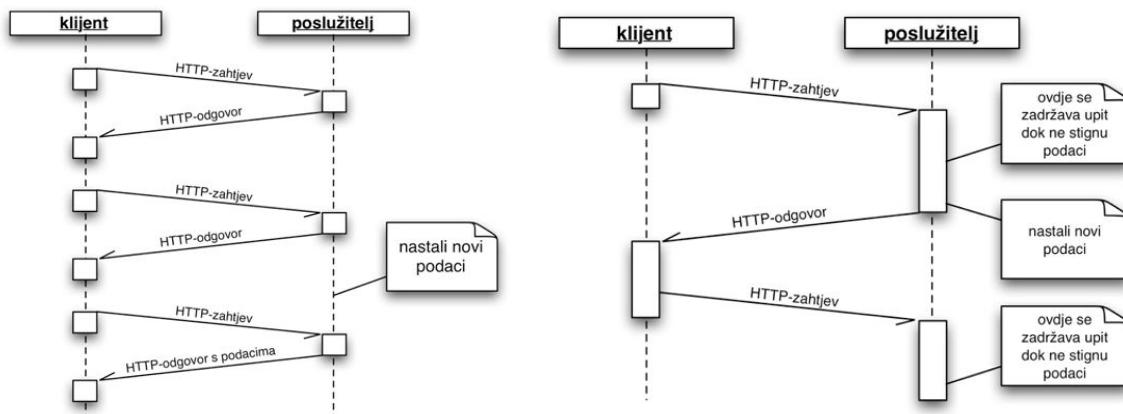
7. Skiciraj i objasni metodu prozivanja poslužitelja i metodu dugog prozivanja posl.

Prozivanje - u slučaju kad nema odgovora, klijent prima prazan odgovor.

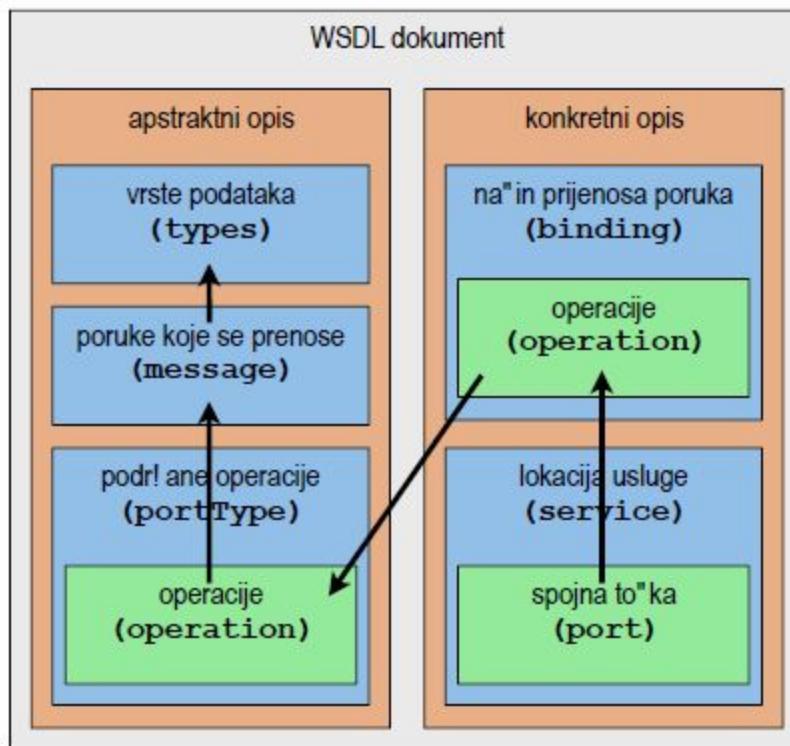
Nedostatak ovog pristupa je što u slučajevima kada nema puno poruka klijent opterećuje mrežu i poslužitelj nepotrebnim zahtjevima.

Dugo prozivanje - klijent šalje upit na poslužitelj, ali mu poslužitelj ne odgovara nego čeka događaj, tj. poruku drugog korisnika, kada poslužitelj šalje odgovor.

Nedostatak je što klijent stalno mora imati otvorenu konekciju prema poslužitelju a poslužitelj mora imati otvoren velik broj konekcija prema klijentima što zauzima resurse.

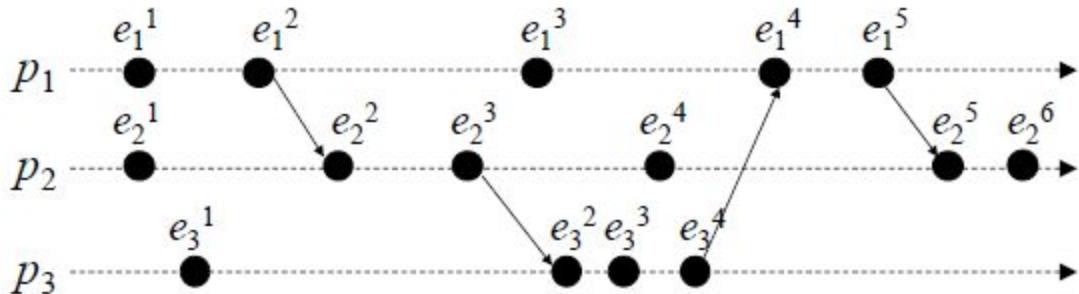


8. WSDL- skiciraj apstraktne i konkretne dijelove, opisati sve.



1. types: definira vrste podataka neovisne o platformi i jeziku (koristi se XML Schema),
2. message: definiraju ulazne i izlazne poruke koje se mogu koristiti kao parametri usluge,
3. operation: predstavlja jednu operaciju/metodu/proceduru koja je definirana u usluzi, a sastoji se od definicija ulaznih, izlaznih i iznimnih poruka koje se mogu razmjenjivati korištenjem ove operacije,
4. portType: koristi poruke (pod 2) kako bi opisao sve operacije koje pruža usluga.
1. binding: definira kako je konkretna implementacija povezana s operacijama u apstraktnom opisu i definira format u kojem će se poruke prenositi (protokol i elemente)
2. service: definira URI na kojem je usluga isporučena tj. na kojoj adresi se može pozvati usluga (taj URI je definiran u spojnoj točki – oznaci port).

9. Uzročno povezani događaji, npr



- a. i b. je bilo jesu li događaji uzročno povezani a c. definiraj kada su događaji konkurentni

10. Navedite i ukratko objasnite dva razloga zbog kojih je potrebno sinkronizirati procese u raspodijeljenoj okolini.

UPORABA DIJELJENIH SREDSTAVA - potrebno je izbjegći istovremeni pristup dijeljenom sredstvu

ALGORITAM ODABIRA VOĐE - neophodno je da procesi budu međusobno sinkronizirani

11.

Prepostavimo da skupina od n procesa implementira algoritam za odabir vođe u sinkronom prstenu. Svaki proces u prstenu odabire vlastiti jedinstveni identifikator UID koji je cijeli broj, procesi su povezani u prstenu na način da imaju jednog prethodnika i sljedbenika, UID nisu sljedbenici u prstenu, a procesi ne znaju n .

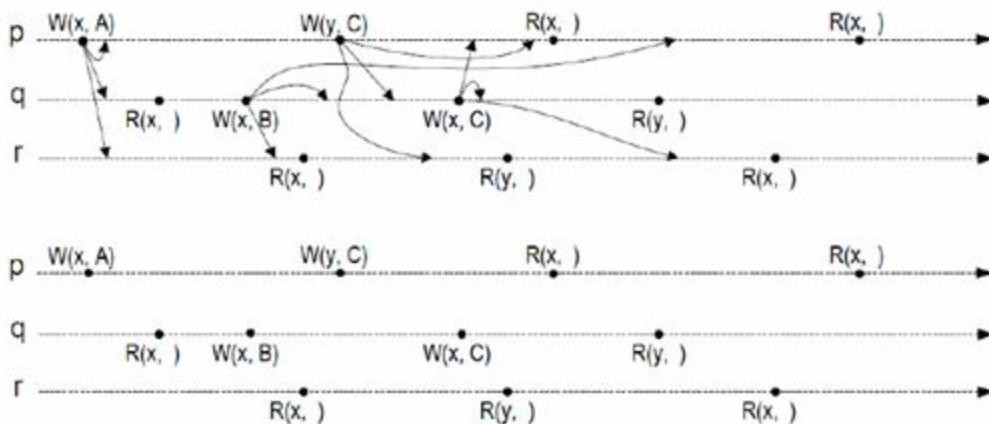
- Opišite ukratko ideju algoritma (traži se skica algoritma, a ne formalni algoritam)
- navedite vremensku i komunikacijsku složenost algoritma.
- Možemo li definirati bolji algoritam u slučaju da procesi znaju n , tj. postoji li kakva prednost pri definiranju algoritma za slučaj da svim procesi znaju n .

a. Svaki proces inicijalno šalje svoj UID susjedu. Kada proces primi UID, ako je taj veći od njegovog UID-a proslijeđuje ga dalje, ako je primljeni UID manji od njegovog UID-a primljeni UID se odbacuje, a ako je primljeni UID jednak njegovom UID-u proces objavljuje sebe kao vođu

b. vremenska: $O(n)$, komunikacijska: $O(n^2)$

12. Ovakav, ali za povezanu konzistentnost

Na slici je prikazan redoslijed izvođenja operacija triju procesa. Objasnite poštije li prikazani slijed izvođenja operacija konzistentnost redoslijeda? Ako da, prikazani primjer promjenite tako da narušite konzistentnost redoslijeda izvođenja operacija. U suprotnom prikazani primjer promjenite tako da ostvarite konzistentnost redoslijeda izvođenja operacija. Obrazložite predloženo rješenje.

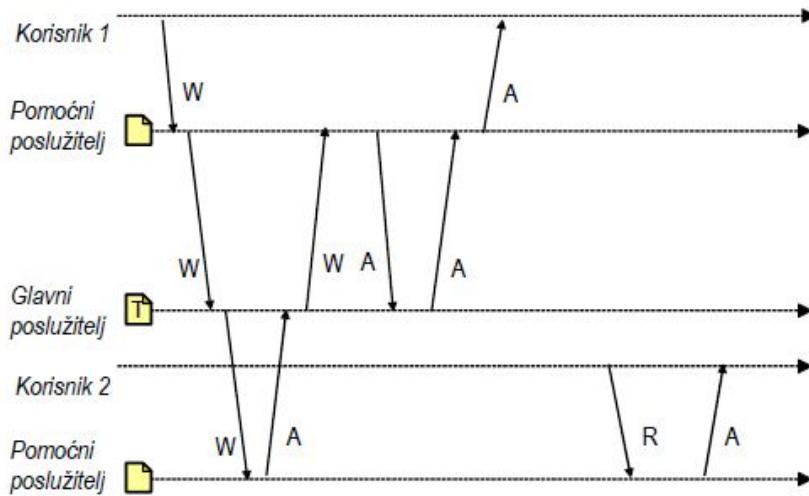


objašnjenje (valjda dobro):

1. korak: upišeš slova na prazna mesta na način da gledaš šta je upisano zadnje, tj. koja je zadnja strelica došla. znači na gornjim procesima, redom ćeš upisati
p: R(x, C), R(x, B)
q: R(y, C)
r: R(x, B), R(y, C), R(x, C)
2. korak: pronađeš povezane operacije, i mislim da je to u slajdu krivo napisano. Operacije su povezane kad proces čita iz lokacije a zatim zapisuje na tu istu lokaciju, kao što je ovdje slučaj kod procesa q. Također, mislim da je povezana i ova operacija W(x, C) na procesu q, jer zapisuje na istu lokaciju. Da zapisuje npr na lokaciju y, onda nebi bile povezane, već konkurentne.
Dakle imamo sljedeće povezane operacije: ABC. Zbog definicije povezane konzistentnosti, procesi p i r moraju povezane operacije vidjeti na isti način.
Gledamo samo čitanje. Proces q: CB → nije dobro, mora ići B pa C. Proces r: BC → dobro je, ali zbog procesa p ipak ne vrijedi pov.konz.

Na drugom dijagramu potrebno je promijeniti strelice tako da se ostvari pov.konz. Dakle, želimo da na procesu p redoslijed operacija bude BC a ne CB. Sve što treba napraviti je pomaknuti dvije strelice, mislim da je sad jasno.

13. Prikazan je prazni slijedni dijagram, potrebno ga je popuniti s koracima udaljenog obnavljanja stanja replike.



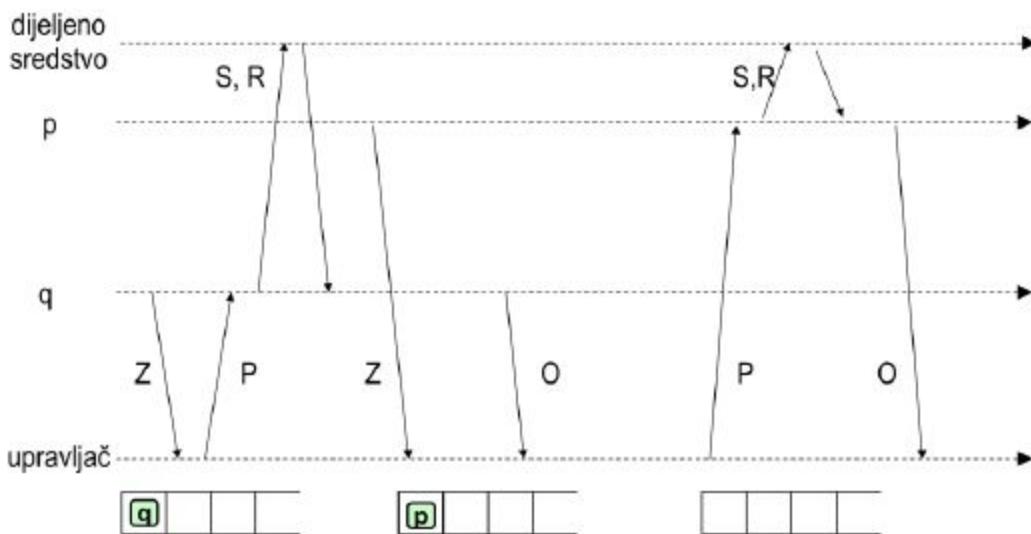
ZI 2015

- 1. Navedi postojeća rješenja za komunikaciju s obilježjima vremenske neovisnosti procesa i neovisnosti o referenci sugovornika. Zašto je navedeno rješenje(a) neovisno o referenci sugovornika?**

To su objavi-preplati i dijeljeni podatkovni prostor. Objavi-preplati je rješenje neovisno o referenci sugovornika jer ne mora znati identifikator pretplatnika, a dijeljeni podatkovni prostor jer je komunikacija anonimna (temelji se na sadržaju podataka).

- 2. Skiciraj i objasni 2 rješenja za međusobno isključivanje procesa kod pristupa dijeljenom sredstvu: središnji upravljač s repom čekanja i međusobno isključivanje primjenom prstena.**

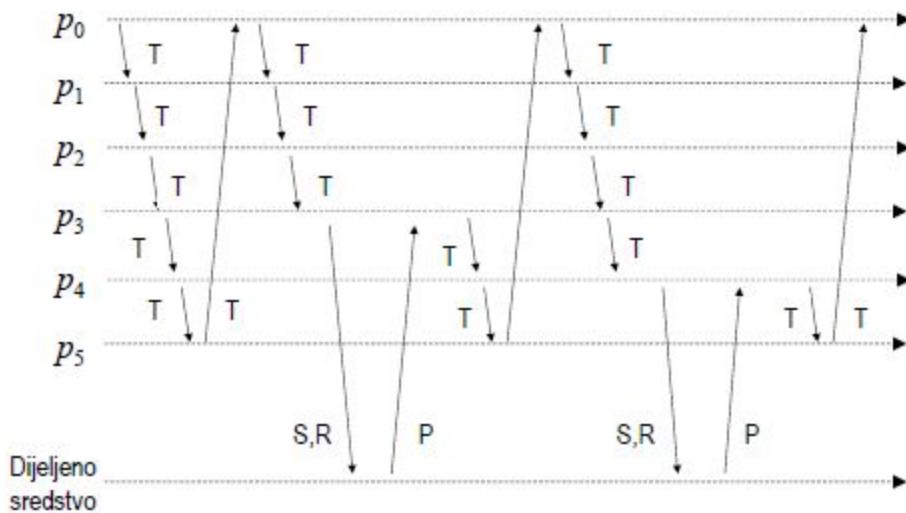
R – Dohvati, S – Spremi, Z – Zauzmi, P – Potvrda, O – Oslobodi



Središnji upravljač s repom čekanja je proces koji upravlja pristupom dijeljenom sredstvu. Pristigle klijentske zahtjeve obrađuje prema redoslijedu prispijeća (FIFO). U slučaju zauzeća dijeljenog sredstva, pristigle klijentske zahtjeve će do oslobađanja sredstva čuvati u svom repu čekanja. Tijek komunikacije:

1. Proces šalje zahtjev za zauzimanjem dijeljenog sredstva (Z) središnjem upravljaču,
2. Proces ostvaruju pristup dijeljenom sredstvu nakon primitka potvrde (P) od strane središnjeg upravljača. Proces obavlja akciju dohvaćanja (R) i/ili spremanja (S) dijeljenog sredstva
3. Nakon završetka obrade, proces otpušta zauzeto sredstvo slanjem poruke oslobođeni (O) središnjem upravljaču.

I – Prijenos tokena, S – Spremi, R – Dohvati, P – Potvrda



Primjenom prstena se omogućava vrlo jednostavno isključivanje procesa u vremenu. Pri tome su procesi povezani u logičku mrežu zasnovanu na prstenu. U sustavu postoji samo jedna značka kojom se ostvaruje pravo pristupa dijeljenom sredstvu te se ona razmjenjuje između procesa u prstenu. Pristup dijeljenom sredstvu ima samo proces koji u određenom trenutku ima značku, a nakon završetka ili u slučaju nezainteresiranosti za pristup dijeljenom sredstvu proces prosljeđuje značku susjednom procesu u prstenu.

1. Čekaj poruku sa značkom od svog lijevog susjeda, a kada ju primiš ostvario si pravo na pristup dijeljenom sredstvu,
2. Ukoliko si zainteresiran za pristup dijeljenom sredstvu provedi operacije pisanja ("spremi") i/ili čitanja ("dohvati") sadržaja i
3. Proslijedi poruku sa značkom desnom susjedu u prstenu.

3. Isti kao 12. u MI

4. Objasni zašto je za pouzdanu komunikaciju skupine procesa važan slijed kojim procesi primaju poruke. Navedi pretpostavke izvođenja atomic multicast-a. I zadani dijagram i pita da li zadovoljava njegova svojstva?

Od velike je važnosti jer utječe na promjene stanja tih procesa.

Prepostavke izvođenja: garantira isporuku poruka u istom redoslijedu svim procesima u grupi uz redoslijed isporuke poruka FIFO od istog procesa; ako se dogodi ispad procesa u trenutku isporuke poruke, poruka se isporučuje ili svim ispravnim procesima ili niti jednom od njih.

5. Izračunaj raspoloživost sustava koji se satoji od 2 paralelna web-poslužitelja (raspoloživost=0,997) i jedne mrežne sklopke (MTBF=1 godina, MTTR=4 sata).

rez: 0, 99953

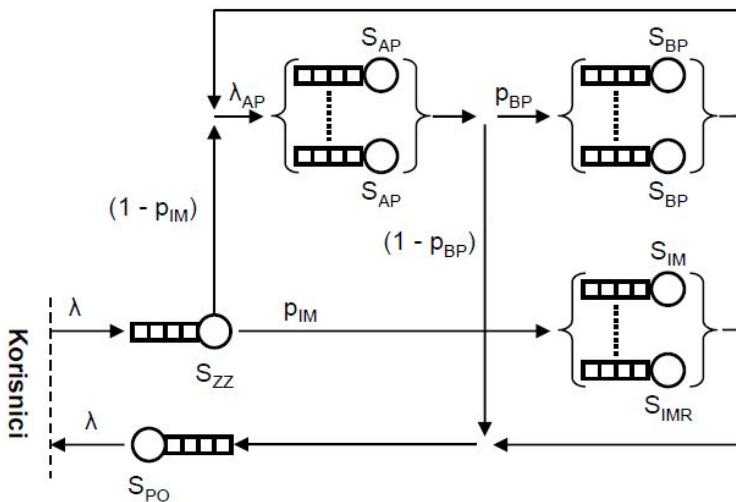
6. Navedi i objasni 2 odabrana nefunkcijska zahtjeva koja smatrati ključnima za kvalitetno pružanje usluga nekog web dućana.

Performanse se definiraju na različite načine i uključuju različite izračunljive i mjerljive parametre koji opisuju sposobnost sustava da ostvari funkcije kojima je namijenjen. Općenito, performanse govore koliko posla sustav može obaviti (kakav je kapacitet) i kako brzo (kakvo je vrijeme odgovora).

Pouzdanost sustava je definirana kao vjerojatnost da sustav radi ispravno u vremenskom periodu t pod definiranim uvjetima okružja. **Raspoloživost sustava** definirana kao vjerojatnost da sustav radi ispravno u trenutku t, odnosno u trenutku kad ga korisnik treba. U praksi se češće rabi neraspoloživost sustava.

7. 2 paralelna podsustava u seriji s još jednim, intenzitet zahtjeva = 0,5 z/sek. $p_1=0,6$ $p_3=0,3$, povratna veza. Vremena posluživanja su 1 s/z, 0,1 s/z, 0,2 s/z. Izračunaj ukupno vrijeme zadržavanja u sustavu.

8. Slika slična ovoj iz slajda. Nap1=3, Nbp=4, Nap2=3. Izraziti p* u ovisnosti o lamda.



9. Zadatak s Chordom kao u zadacima za vježbu.

10. Nacrtaj slojeve u spletu računala.

korisničke aplikacije

zajedničke usluge (collective)

protokoli za sredstva (resource)

komunikacijski protokoli (connectivity)

osnovna sredstva (fabric)

11. Objasni zašto je cloud prikladan za razvoj aplikacija u području IoT.

imamo neke senzore koji su ubiti te "stvari", oni prikupljaju i razmjenjuju podatke preko neke mreže s prilazom tj. mobitelom koji to preko interneta šalje u računalni oblak na obradu

2 tipa obrade podataka: obrada toka podataka u stvarnom vremenu i big data analytics

12. Objasni što je stvar i navedi obilježja stvari koje su povezane na internet.

- Objekt iz fizičkog svijeta (fizički objekt ili stvar, npr. senzori i aktuatori) ili virtualnog digitalnog svijeta (virtualni objekt)
- Internet Connected Object (ICO)
 - ima jedinstveni identifikator i povezan je na Internet
 - komunicira i kontinuirano generira podatke
 - može primati podatke, naredbe za konfiguraciju
 - može primati podatke od drugih ICO, obradivati ih i slati dalje na obradu u računalni oblak

Internet of Things (IoT): dobra svojstva



- Postoji jedinstven identifikator za svaku „stvar“
- Primjena interoperabilnih komunikacijskih protokola
- „Stvari“ su integrirane u informacijsku mrežu
- Jednostavno pronađenje i samokonfiguracija stvari
- Dinamična i prilagodljiva rješenja

13. Objasni razliku između razmernog rasta primjenom DNSa i primjenom mrežnog rutera.

Raspoređivanje na DNS poslužitelju- Grozd računala ima jedno ime. Na temelju zadanog imena, primjenom DNS-a ostvaruje se razlučivanje imena na strani poslužiteljskog DNS poslužitelja (on sadrži više adresa grozda računala i odabire određeno računalo te IP adresu odabranog računala vraća kao rezultat). Nakon razlučivanja imena, korisnik proslijeđuje zahtjev na odabrano pristupno računalo. Prednost je da korisnička strana ne mora znati za raspodijeljenost grozda, nego ga koristi kao da je to jedno računalo. Nedostatak je održavanje zapisa u DNS poslužitelju koje uzrokuje da jedan klijent neko vrijeme (dok ne istekne valjanost razlučivanja) koristi jedno poslužiteljsko računalo.

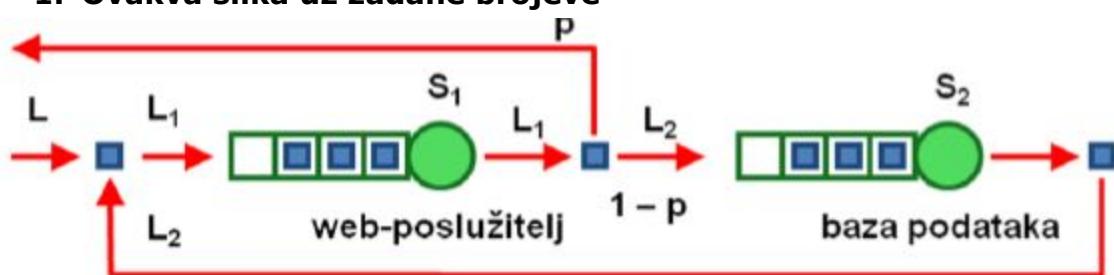
Raspoređivanje na mrežnom usmjeritelju- Grozd računala i ovdje ima jedno ime. U DNS sustavu za to ime prijavljena je samo jedna adresa, adresa mrežnog usmjeritelja koji je pristupna točka grozdu računala. Nakon razlučivanja imena, korisnik proslijeđuje zahtjev mrežnom usmjeritelju. Mrežni usmjeritelj sadrži popis

svih adresa web-poslužitelja grozda. On odabire jedan web-poslužitelj i prosljeđuje mu zahtjev. Za sljedeći zahtjev mrežni usmjeritelj odabire neki drugi web-poslužitelj. Postupak odabira može biti zasnovan na različitim disciplinama posluživanja.

Prednost- korisnički sustav ne razlikuje korištenje grozda od korištenja jednog poslužitelja. Nedostatak- uporaba i održavanje zapisa u mrežnom usmjeritelju kao i sav promet koji prolazi kroz njega prema svim klijentima.

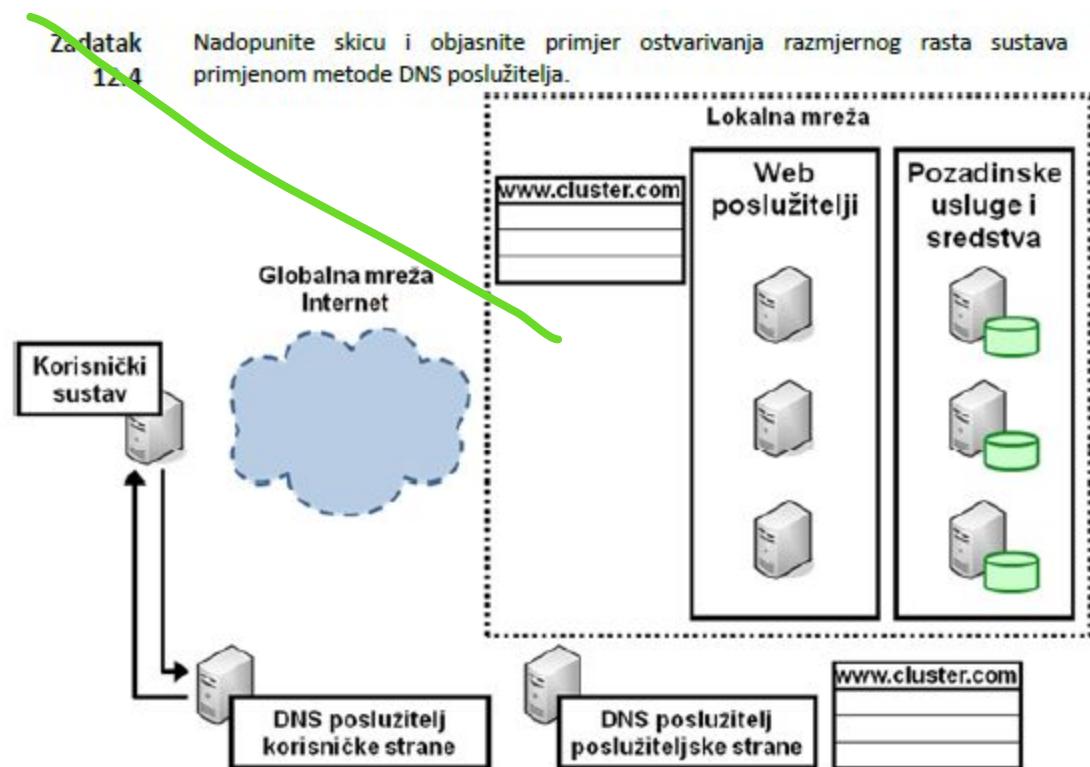
ZIMSKI ROK 2015

1. Ovakva slika uz zadane brojeve



Slika 10.23. Model web-usluge

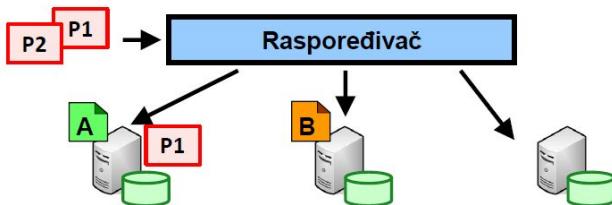
2. Nadopuniti sliku



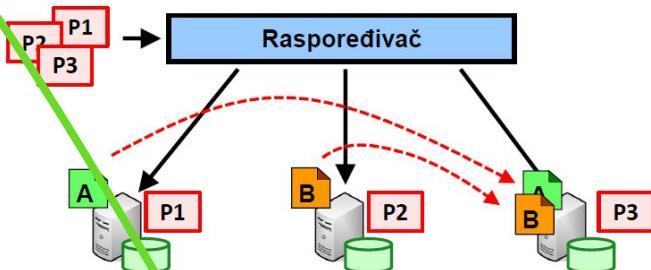
LJETNI ROK 2015 - nova pitanja

1. Objasnite razliku između prostorne i vremenske lokalnosti.

- ◆ Primjer korištenja prostorne lokalnosti



- ◆ Primjer korištenja vremenske lokalnosti



- ◆ Prostorna lokalnost

- Poslovi se raspoređuju na čvorove koji sadrže podatke potrebne za izvođenje posla
- Poslovi se približavaju podacima

- ◆ Vremenska lokalnost

- Rezultati izvođenja poslova postavljaju se na računalo gdje se izvode poslovi koji koriste dobivene rezultate
- Podaci se približavaju poslovima

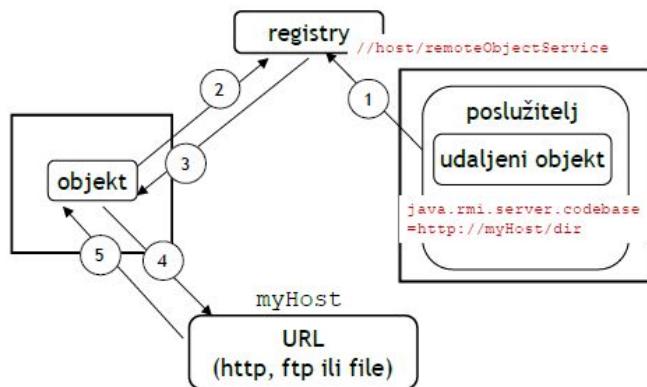
2. Koja je uloga skripte na klijentu?

poglavlje 4.7 u knjizi

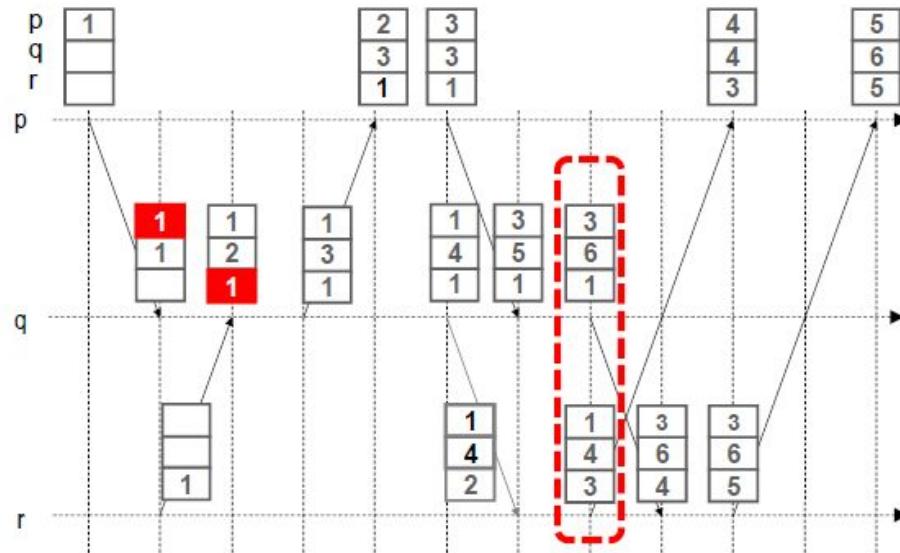
3. Koji su izazovi prisutni kod IoT?

slide 19, zadnja prez

4. Bila je zadana ova slika, trebalo je napisati i opisati korake te navesti u kojem koraku se događa serijalizacija parametara.

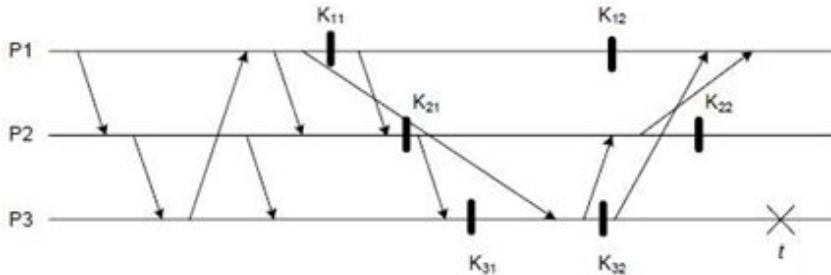


5. Vektorske oznake vremena, bila je zadana sljedeća slika ali s praznim kućicama, popuniti ih i opisati.



6. Ovaj zadatak iz zadataka za vježbu

Slika prikazuje tri procesa i razmjenu poruka među njima. Svaki proces neovisno o drugim procesima bilježi svoja stanja u označenim kontrolnim točkama. U trenutku t dolazi do ispada procesa P3. Možemo li sustav od tri procesa na slici nakon ispada procesa P3 vratiti u konzistentno stanje koristeći kontrolne točke K_{11} , K_{21} i K_{31} i objasnite zašto je to moguće ili nije moguće?



7. 3 osnovne razlike između web-aplikacija i web-usluga

8. Navesti i objasniti 3 svojstva metoda HTTP protokola.

sigurnost -..

idempotentnost- ..

cachable - ..

9. Odaberi među ponuđenim ona obilježja koja upisuju dijeljeni podatkovni prostor. Koje od navedenih obilježja nije svojstveno za komunikaciju porukama? Objasnite.

vremenska ovisnost / vremenska neovisnost

ovisnost o referenci sugovornika / neovisnost o referenci sugovornika
perzistentna / tranzijentna

sinkrona / asinkrona

push / pull

Kod komunikacije porukama, postoji ovisnost o referenci sugovornika, jer pošiljatelj mora znati identifikator odredišta tj repa gdje će spremiti poruku

10. neka ogromna tablica, po recima su bili operacije, raspoređivanje, doseg itd a po stupcima baze podataka, WWW i HTTP. i sad je trebalo popuniti tu tablicu..

11. Mehanizam semafora, nadopuniti skicu

12. pomoću vennovih dijagrama prikazati odnos konzistentnosti

ZI 2013/2014

1. onaj zadatak sa 3k+1, di vam je bio zadan k

to je 8.1 iz oglednog, k su procesi u stanju bizantskog ispada, a n je min broj procesa u grupi za postizanje sporazuma

2. Utječe li komunikacijska složenost na skalabilnost, i neke 2 stvari

Komunikacijska složenost algoritma je važna za skalabilnost raspodijeljenog

sustava jer na temelju komunikacijske složenosti možemo zaključiti kako raste generirani promet raspodijeljenog sustava s rastom tog sustava. Primjer: komunikacija grupe procesa.

3. Grozd vs splet vs cloud arhitektura, infrastruktura

prez 12, slide 6

4. strukturirani vs nestrukturirani P2P.

Kod nestrukturiranih je podtaka na jednom sudioniku, i može biti na drugima. Nestrukt. se točno zna koji sudionik ima koji podatak, i zato je pretraga logn, što ih čini skalabilnima.kod strukturiranih se zna tko ima podatak i strukt su skalabilni jer se pretraživanje odvija u log(n) koraka

5. TCP, odrediti redoslijed naredbi za klijenta i servera

(ono socket() connect() bind...)

1.ciklus

6. Dvoslojna arh, ilustrirati i u jednoj rečenici opisati.

Tanki i debeli klijent, tanki ima logiku na udaljenom računalu, a lokalno samo prikazuje dobivene podatke.

7. Zadatak sa Chordom, opisati 5 koraka algoritma

ako se misli na dodavanje peera:

- a. novi čvor, koristeći funkciju lookup, pronađi svog sljedbenika
- b. novi čvor postavlja pokazivač na sljedbenika i od njega saznaće prethodnika
- c. čvor koji je bio sljedbenik mijenja svog prethodnika, a onaj koji je bio prethodnik, mijenja svog sljedbenika
- d. podešava se tablica usmjeravanja za novi čvor
- e. podešavaju se tablice usmjeravanja za sljedbenika i prethodnika

8. Ono neko slanje poruka m i vc. m se počne slati, a pošiljatelj otpadne, šta se desi?

Oni koji su dobili poruku m je ignoriraju, a ostali je ni ne prime.

9. Zamislite da imate neka 4 procesa stroge konzistentnosti. Jesu li oni nužno i povezane konzistenosti?

Strogi kažu da svi procesi moraju vidjeti sve u pravom redoslijedu. Povezana kaže da samo povezani procesi moraju vidjeti sve u pravom redoslijedu. svi je podskup od neki XD

10. 3 nefunkcijska svojstva.

performanse, pouzdanost/raspoloživost, ukupni trošak vlasništva (skupno se zovu kvaliteta usluge)

11. razlika između pouzdaonsti i raspoloživosti; trivijalni zadatak sa 2 servera i routerom.

prez 9, slide 23, 24, 25

12. Zadatak sa checkpointovima, pita šta je sa točkom A,B,C

Jedna je izgubljena, ali će se ponovno poslat, druga će se ponovno poslati zbog checkpointa, a 3. će doći tek nakon undo-a.

14. Imate L koji se razdvaja na 2 repa i procesa i na kraju se opet spaja u jednog. Vjerojatnost da L ode u jednom smjeru je p, u drugom smjeru je (1-p), izacunat razne gluposti

15. Još jedan takav zad. Imate zadan L, trebate naći L1, L2, L3 i Liz. Sustav je serijski sa 3 repa za redom, s tim da je nakon svakoga strelica koja izlazi iz sustava i ima 30% šanse da se desi.

ZIMSKI ROK 2014

- Koju bi međuopremu koristili za realizirati sustav praćenja tramvaja u realnom vremenu**

IoT

Poll, long poll - skicirati i objasniti

- Opisati skalabilnost, otpornost na kvarove, ...
- Lokacijska, relokacijska, migracijska transparentnost - opisati u kontekstu weba**

Lokacijska: prikrivanje lokacije sredstva, npr korisnik zna simboličko ime poslužitelja i DNS će razlučiti to u IP adresu, no lokacija poslužitelja (IP adresa) korisniku ne mora biti poznata

Relokacijska: dostup sredstvu i prilikom njegova premještanja. Ova transp. na webu nije prisutna jer je web-poslužitelj stacionaran. Relokacijska je vidljiva npr kod mobile IP.

Migracijska: prikrivanje promjene lokacije sredstva: može se promijeniti IP adresa (lokacija poslužitelja) bez da se mijenja simboličko ime

- Raspoređivanje zahtjeva na strani klijenta**

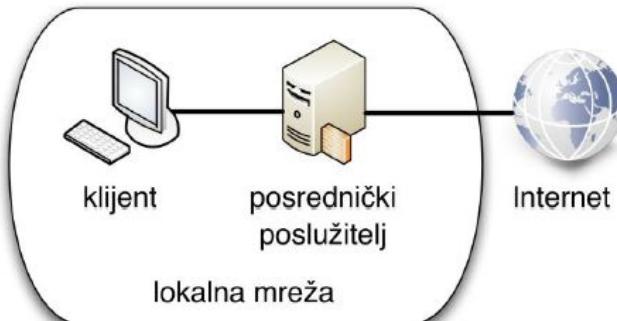
Korinički sustav sadrži adrese grozda računala, odabire jednu adresu i šalje zahtjev odabranom računalu. Sljedeći put šalje zahtjev na iduću adresu tj iduće pristupno računalo itd.

- Računski zadatak s modelom repa čekanja s povratnim vezama (web-poslužitelj i baza podataka)**
- Računski zadatak, izračunati a) PULL i b) PUSH uz obavijesti o promjenama**
- Dijagram sa vektorskim oznakama vremena**
- Oporavak povratkom unazad**
- Tri tipa cloud usluga**

(ne znam je li se mislilo na IaaS/PaaS/SaaS ili private/public/hybrid cloud)

- Skicirati proxy na strani klijenta (forward proxy) i objasniti dva glavna razloga zašto je koristan**

Kada se posrednički poslužitelj nalazi u lokalnoj mreži klijenta (engl. *forward proxy*) onda se smanjuje korištenje veze između lokalne mreže i ostatka mreže što radi uštedu organizaciji (Slika 4.5). Organizacija može filtrirati i kontrolirati sadržaj pomoću ovakvog posredničkog poslužitelja.



Slika 4.5. Posrednički poslužitelj u lokalnoj mreži klijenta

- **Zadatak objavi-preplati s algoritmom preplavljanja preplatama**
- **Dana je skica odgođenog sinkronog RPC-a bez oznaka, pitanje je što skica predstavlja i objasniti jedan način primjene**
- **Objasniti karakteristike i razliku između centraliziranih i decentraliziranih raspodijeljenih sustava**
- **Dana su dva isječka XML-a, pitanje je što predstavlja (odg: SOAP zahtjev i SOAP odgovor)**
- **Nadopuniti dijagram - primjer sinkronizacije tijeka izvođenja procesa putem mehanizma semafora**
- **Što je ispad kanala a što ispad procesa; koje su dvije glavne vrste ispada procesa**

ispad kanala: proces p šalje poruku procesu q, ali proces q poruku ne prima

ispad procesa: bizantski - proces generira proizvoljne izlaze

-proces ne mijenja stanje iako se ne nalazi u završnom stanju

- **Nadopuniti dijagram s metodom lokalnog obnavljanja stanja replike**
- **Što je konzistentnost redoslijeda; za slučaj s primjera vrijedi li konzistentnost redoslijeda; ako da, narušiti ju, ako ne, ostvariti ju**
- **Chord mreža - kao u primjerima, samo s drukčijim podacima - izračunati dvije routing tablice, gdje će novi podatak biti zapisan, redoslijed čvorova, što će se promijeniti u routing tablici ako dodamo čvor**
- **Zadana je matematička formula (iz skripte) uzročne povezanosti s viticom od tri elementa, pitanje je što predstavlja**

$$e_i^x \rightarrow e_j^y \Leftrightarrow \begin{cases} e_i^x \rightarrow e_j^y, (i = j) \wedge (x < y) \\ e_i^x \rightarrow_{msg} e_j^y \\ e_i^x \rightarrow e_k^z \wedge e_k^z \rightarrow e_j^y \end{cases}$$

- a) događaji su uzročno povezani ako su jedan iza drugog na istom procesu; b) ako postoji slanje i primanje poruka između njih;
 c) definira tranzitivnu uzročnost

ZIMSKI ROK

- 1) Bio je zadatak iz zadace -> izracunat vrijeme odziva u dva slučaja (kada se poveća intenzitet dolazaka i broj operatera i kada se smanji vrijeme posluživanja sa 3 na 1.5min)**
- 2) Shema povezanih sustava, napisat analitičke izraze za intenzitet dolazaka, zadrzavanje u sustavu i još nesto.**
- 3) Objavi-preplati**
- 4) Koje modele podržava JMS (objavi-preplati i komunikaciju porukama)**
- 5) Migracijska, lokacijska i relokacijska transparentnost**
- 6) Chord struktura, tipičan zadatak ako se ne varam**
- 7) Prozivanje i dugo prozivanje, nacrtat i objasnit**
- 8) Uloga sloja raspodijeljenog sustava**
- 9) Skicirat pomocni posluzitelj u lokalnoj mrezi posluzitelja i objasnit prednosti**
- 10) Na primjeru HTML zahtjeva napisat obavezne dijelove i objasnit ih**
- 11) Odgodeni sinkroni poziv RPC**
- 12) Slika sa tri procesa, naci zavisnosti, vrijedi li Casual Ordering i jel FIFO**
- 13) Skica i model rada skupine procesa u kojima postoji koordinator i ostalo su radne dretve**
- 14) Objasniti primjer razmjernega rasta sustava preko metode DNS posluzitelja (dvije metode, nemrem se točno sjetiti)**
- 15) Sto je konzistentnost u sustavu i nacini upravljanja konzistentnosti replika u sustavu**
 udaljeno i lokalno obnavljanje stanja replike
- 16) Iskljucivanje dva procesa primjenom srednjeg upravlјaca sa repom cekanja**
- 17) Razlika između Cristianovog i Berkley algoritma i kojoj vrsti sata pripada svaki od njih (oba su fizicki sat)**
- 18) Razlika ispada procesa i kanala i nabrojiti dva tipa ispada procesa**
- 19) Nacrtat strelice za lokalno obnavljanje stanja replike**

20) Konzistentnost redoslijeda (dal slika zadovoljava, ak da sruši konzistentnost, ak ne postigni konzistentnost)

21) Izracunat opterecenja kod metoda: pull i push sa proslijedivanjem obavijesti o promjenama

22) Koja dva nefukcionalna svojstva trazite kod web-shopova, objasni zasto

23) Navedi barem dva razloga zasto dolazi do razlike u vremenima razlicitih procesa

JOS NEKA PITANJA S FER2

- Zadnja trećina gradiva:
 - Iz 10. poglavlja su bila ona ista dva koji su bili i na roku (jedan s povratnom vezom i onaj s paralelnim posluziteljima).
 - 2 zadatka s trazilicom - jedan teoretski i jedan s racunanjem (vraga mi se to rjesavalo)
 - Teorija sirenja preko DNSa odnosno routera
 - Nacrtati i objasniti djelove clustera
 - Napisati tok poruka za 2-phase committ
 - Ostalo sve vec vidjeno - dosta teorije, par laganih zadacica
 - bila je ona za formula uvjetne dogadjaje, konzistentnost,
 - lokalno obnavljanje replike,
 - teorija fizickih satova,
 - ispadi kanala / procesa (i napisati 2 tipa ispada procesa)
 - cemu sluzi open proxy u javnoj mrezi
 - Cemu sluzi sloj raspodijeljenog sustava
 - Tablica s sigurnosti, idenpotentnosti i cachable za HTTP naredbe
 - Napisati i objasniti one 3 naredbe s tupleovima (write, read i jos neka, nisam ju znao)
 - 3 tocke sinkronizacije pošiljatelja i primatelja
 - Nacrtaj i objasni Pull i Long Pull kod WS
 - Nacrtaj usluge weba (onaj zuti graf)
 - Podsistavi grozda
 - U kakvoj je vezi inverzni indeks i rječnik kod raspodijeljenih trazilica
 - Odredjivanje vodje kod asinkronog prstena
 - Racunanje slicnosti dokumenata kod neceg, ono s itf i log n/df
1. U raspodijeljenom sustavu se komunikacijska međuoprema koja implementira model komunikacije objavi-preplati koristi za (što, lista)
 2. Kako se izvodi prirodni model tereta prilikom vrednovanja nefunkcijskih obilježja ras. sus.

3. Koje biste parametre promatrali u slučaju web aplikacije za e-trgovinu koja na temelju kratkih korisničkih uputa dohvaća informacije o proizvodima u dva oblika - posebni kratki dokument ili katalog u većem dokumentu?

USMENI 2015

P2p mreže, vrste, primjeri, kako rade
ispadi, vrste, primjeri detekcije..pokazao sam mu na primjeru kad se dogovaraju
što je transparentnost, primjeri..

Navedi koje sve modele komunikacije poznajes

Navedi sve znacajke p2p sustava i njihove primjere i primjene.

Objasni sto je konzistentnost i kako se ostvaruje (ne konzistentnost replika, nego
prava konzistentnost - pojma koji označava da je sustav u poznatom stanju i da svi
korisnici vide to poznato stanje)

Za apsolutno sve je pitao nesto stila "Kad biste u stvarnom životu imali prijatelja
koji bi radio xy sustav, kakav biste mu model preporučili i zasto"

Vrste ispada i opisi svaki. Kako se otkrivaju i sporazumi kod bizantskih. Kontrolne
tocke i dnevnički zapisi.

Replike i konzistentnost replika, kakve replike postoje (korisničke, poslužiteljske i
trajne), nacrtati poslužitelj s repom i pristup bazi (koja isto ima rep, kao na
slajdovima, zadatak iz zadaće)

- otvorenost u RASUS-u, neki primjer zašto je to važno u telekomunikaciji
- vremenska ovisnost i neovisnost kod slanja poruka
- standardi kod web usluga
- zašto su važne performanse kod sustava, što bi napravio da imam neki web shop i
da moram poboljšati performanse
- razlika između spletova i grozda
- kako bi poboljšao performanse grozda

Pitao malo o transparentnostima, o grozdu i spletu, o replikacijama

Pitala me P2P sustave, kakvi postoje i koja je prednost tijedan nedostatak svakog. Nešto
malo o vektorskim označama. Kolege prije mene je pitala skalabilnost i vertikalno i
horizontalno skaliranje.

Mene je pitala o konzistentnostima, replikama, obnova nakon ispada - ima hrpu
slajdova i onda ti da neki i malo objasniš, kako ugodan usmeni
pitanja su bila o razmernom rastu, lokalna vs vremenska raspodjela poslova i
podataka usmjerivačem, three phase commitom i zašto je bolji od 2 phase, pa malo
kako se sve može koordinirati procese, tj. s centralnim koordinatorom, biranjem
vođe ili ravnopravno itd.

Hijerarhijska i ravnopravna organizacija peerova.

Zajednicki memorjski prostor.

MM1 model, cemu služi, sta koje slovo znači itak o njemu općenito

otvorenost, što je to i da li je bitno u telekomunikacijskom sustavu, zatim me pitao što je poll, a što long poll i što se time postiže, i na kraju me pitao da radim u ministarstvu znanosti i da mi kazu da trebam za neki projekt super jako računalo, kako bih pristupio tome (dakako odgovor je bio u izgradnji grozda).

USMENI

Kušek- na ispitivanju ima isprintane slajdove i izvlaci se po jedan iz 3 kategorije, uglavnom treba objasnit sta je na slajdu i dat neki primjer(meni je bio nacrtan komunikacijski kanal, asinkrono stablo i sklopka za rasporedjivanje zahtjeva)

1)centralizirani sustav objavi/preplati

2)usmjereni sinkroni graf

3)littleov zakon

Vrste komunikacije (konekcijska, beskonekcijska, tranzijetna i perzistentna), vrste web-usluga (temeljene na RCP-u, porukama i prikazu stanja resursa) i nedostaci kod računarstva u oblaku.

Kod Kušeka TCP socket primjer TCP klijenta (prezentacija 2a slajd 34 i 35) treba reći nešto o tokovima podatak piše u knjizi nisam znao , web-usluge slojevi(prezentacije, poslovne logike, obrade podataka) ona slika kada klijent koristi uslugu (prezentacija 3. slajd 66) i 3 phase commit (prezentacija 8. slajd 43.

Bio sam kod Pripužića u 11h. Istina je sve što kažu u prošlim temama. Ugodan je za razgovor i blag. Ima isprintane prezentacije ispred sebe i prvo pitanje mi je postavio o transparentnostima. Otvorio je slajd na kojem su nabrojane sve transparentnosti i rekao mi je da kažem po jednu rečenicu za svaku od njih. Nakon toga me pitao razliku između grozda i spleteta. I onda je izvukao prezentaciju o sinkronizaciji procesa u vremenu i pitao me koji su problemi pri sinkronizaciji procesa. Nedostatke fizičkih satova; logičkih i vektorskih oznaka i morao sam mu na primjeru sa slajda objasniti princip rada vektorskih oznaka. Dao mi je 9/10. Usmeni kod Pripužća, mrcvario me dobrih 25 minuta. Pita 3 pitanja s time da je prvo iz starog gradiva (prvi ciklus) tako da oprez. Mene je pito bio odaziv i preciznost za tražilice, koje nismo radili, ali mi je to glupo bilo reći jer nisam bio sto posto siguran da nismo i jer nisam bio ni na jednom predavanju.

3 načina komunikacije u modelu objavi preplati: preplavljivanje preplatama, obavjestima i neko filtriranje nisam ga baš najbolje skonto, to smo kao na predavanju radili i bome me tu mrcvario koje su prednosti, a koje mane tog trećeg načina

semafori

međusobno isključivanje procesa: središnji upravljač i rasподijeljeno međusobno isključivanje

pretraživanje u p2p: strukturirani vs nestrukturirani

tražilica: odziv i preciznost

2. pitanje sa ispita: čemu služi valjda model objavi pretplati il tak neš

Podnar-Žarko- ee da jedan mali hint -> obavezno pita sljedeće : kod slanja zahtjeva što se prvo radi? - razlučuje se IP adresa pomoću DNS-a ..

Ja sam bio kod Podnar-Žarko, ok je bila. Ispituje uz pomoć isprintanih slajdova bez teksta, gdje možda fale neke stvari na slici. Izvuče random slajd i onda te pita malo da objasniš.

Mene je pitala RMI (remote method invocation), da malo objasnim koja je svrha i kako se izvodi. Iz novog gradiva me pitala što su replike, koje su vrste i kad se koriste, i ono kad usmjerivač usmjeruje korisničke upite na trajne replike.

Moja pitanja:

- slajd 7, predavanje 5. - što prikazuje slika, što je uzročna ovisnost i tako to...

-Idealni paralelizam i Amdahlov zakon - slajdovi 43., 44., 45. - 9. predavanje.

RMI, 2pc i 3pc

oporavak od pogreške i trajne datoteke na grozdu

Podnar: nakon sto sam citao dojmove usao sam splasen kao ovca pred klanje. Na kraju dobio 8 bodova iako nisam briljirao.

Pitala me racunarstvo u oblaku, prednosti i nedostatke, sto je elasticnost kod njih (nikad cuo), i jos nesto al se vise ne mogu sjetiti :) al sve u vezi oblaka.

Ono sto nisam znao me navlacija da se sjetim, sto potpitanjima, sto pricom. Uopće nije ostavila dojam babaroge, dapace, rekla mi je da se smirim i samo polako.

Jedino sto jest cinjenica, je da voli da joj se "uceno" odgovara, pa ak nesto opisuje svojim rijecima, ona to ponovi na ljepsi nacin.

Pitala me otpornost na ispadne, REST kod HTTPa i uzročnu ovisnost događaja.

Pitala procese i K.O. (kako ih sync), chord

[naziv tvrtke]

[naslov dokumenta]

[podnaslov dokumenta]

Sadržaj

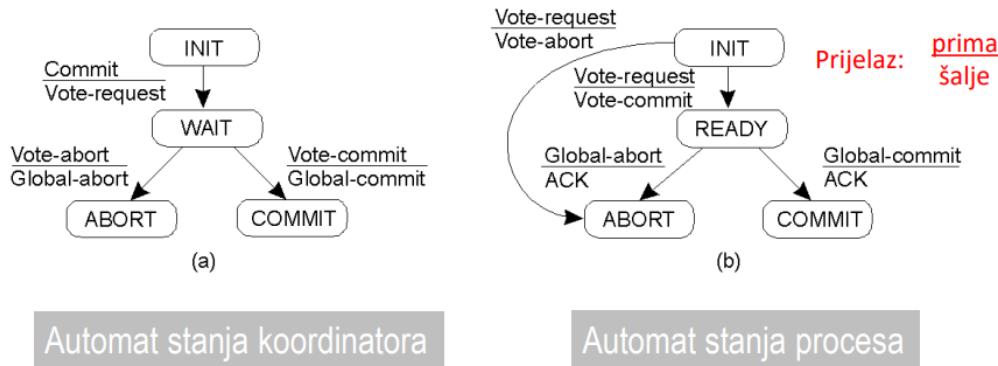
7. Otpornost na neispravnosti u raspodijeljenom okružju	2
8. Računarstvo u oblaku i performanse.....	4
9. Mikrousluge (mikroservisi), komunikacija temeljena na događajima	11
10. Replikacija podataka	14
11. Particioniranje i konzistentnost podataka	16
12. Sustavi s ravnopravnim sudionicima	18
13. Tehnologija raspodijeljene glavne knjige	19

7. Otpornost na neispravnosti u raspodijeljenom okružju

Prepostavite da postoji skupina od m jednakih procesa. Navedite toleranciju ove skupine procesa na ispade ako prepostavite „obični“ ispad ili bizantski ispad.

Tolerancija je osigurana s $k+1$ procesa za ispad k procesa.

Identificirajte i objasnite blokirajuća stanja protokola 2PC.



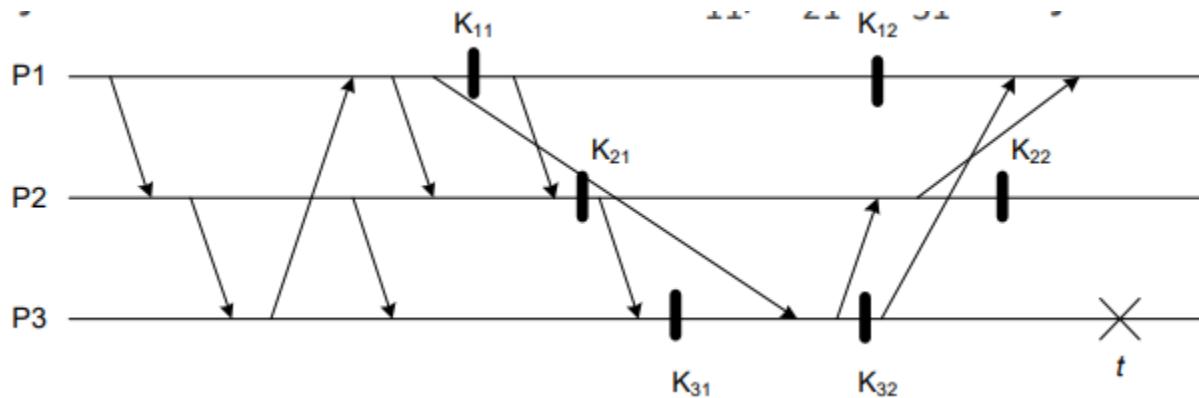
Proces je blokiran u stanju INIT kada čeka VOTE_REQUEST, ako ne primi poruku nakon određenog vremena, proces može lokalno odustati od izvršavanja operacije.

Proces je blokiran u stanju READY čekajući konačnu odluku koordinatora, proces treba sazнати коју је поруку координатор послao и питати друге процесе што се догађа.

Koordinator je blokiran u stanju WAIT kada чека одговор свих процеса, ако након неког периода не прими одговор од свих процеса, координатор може закључити да треба одустати од изvršavanja operacije и poslati свим процесима GLOBAL_ABORT

2PC je blokirajući protokol, jer u slučaju ispada koordinatora nakon slanja VOTE_REQUEST, процеси не могу закључити о sljedećoj operaciji коју требају првести (svi su u stanju READY)!

Slika prikazuje tri procesa i razmjenu poruka među njima. Svaki proces neovisno o drugi procesima bilježi svoja stanja u označenim kontrolnim točkama. U trenutku t dolazi do ispada procesa P3. Možemo li sustav od tri procesa na slici nakon ispada procesa P3 vratiti u konzistentno stanje koristeći kontrolne točke K11, K21 i K31 i objasnite zašto je to moguće ili nije moguće?



8. Računarstvo u oblaku i performanse

Koja je razlika između HPC-a (High-Performance Computing) i HTC-a (High-Throughput Computing)?

HTC radi kontinuiranu obradu velike količine raznovrsnih i nepovezanih podataka dok HPC radi paralelnu obradu velike količine sličnih i povezanih zadataka u kratkom periodu.

Što je grozd?

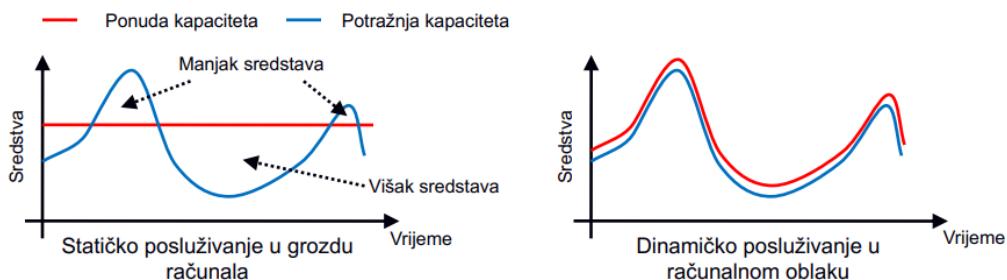
Skup samostalnih računala koje rade kooperativno i čine jedan jedinstveni računalni resurs. Računala su međusobno povezana lokalnom mrežom.

Što je splet?

Skup međusobno udaljenih heterogenih računalnih sredstava koje su međusobno (slabo) povezana Internetom. Rade na zajedničkom zadatku radi postizanja zajedničkog cilja.

Što je oblak?

Uslužno računarstvo, računalni resursi se iznajmljuju po potrebi i te resurse plaćamo onoliko koliko se koriste („pay as you go“). Prividno postoje neograničeni računalni resursi.



Usporedite grozd, splet i oblak po arhitekturi, resursima i aplikacijama.

	Grozd (Cluster)	Splet (Grid)	Oblak (Cloud)
Arhitektura	Skup računala povezanih brzom lokalnom mrežom	Skup udaljenih računala (ili grozdova) povezanih putem Interneta	Virtualizirani grozd računala koja se nalaze u jednom ili više podatkovnih centara
Resursi	Identična ili vrlo slična računala	Raznolika računala	Identična ili vrlo slična računala
Aplikacije	HPC, tražilice	Raspodijeljeno rješavanje problema	Uslužno računarstvo (<i>utility computing</i>), pohrana podataka
Primjeri	Google search engine, Cray XK7, BlueGene/Q, itd.	Folding@home, BOINC, SETI@home, itd.	Microsoft Azure, Amazon EC2, Google App Engine, itd.

Koja je razlika između iznajmljivanju resursa u oblaku i kada kupujemo vlastita računala?

U oblaku možemo jednostavno dodavati i smanjivati resurse ovisno o našoj potrebi, kod vlastitog računala u startu moramo odrediti koje ćemo komponente koristiti i nadogradnja zahtjeva kupnju i zamjenu komponenti. O vlastitom računalu brinemo sami (OS, storage, konfiguracija) dok u oblaku, ovisno o tipu usluge, o tome brine pružatelj usluge.

Koje su vrste oblaka po smještaju i svrsi?

Javni oblik (public) - iznajmljuje se za javnu uporabu i u vlasništvu je organizacije koja prodaje usluge u oblaku.

Privatni oblik (private) - u privatnom vlasništvu poduzeća i samo to poduzeće ga koristi, ne smatra pravim „oblakom“.

Zajednički oblik (community) - nekoliko organizacija dijeli jednu infrastrukturu.

Hibridni oblik (hybrid) - kompozicija 2 ili više oblaka različitih vrsta.

Koja su dva osnovna koncepta u računarstvu u oblaku te ih objasnite?

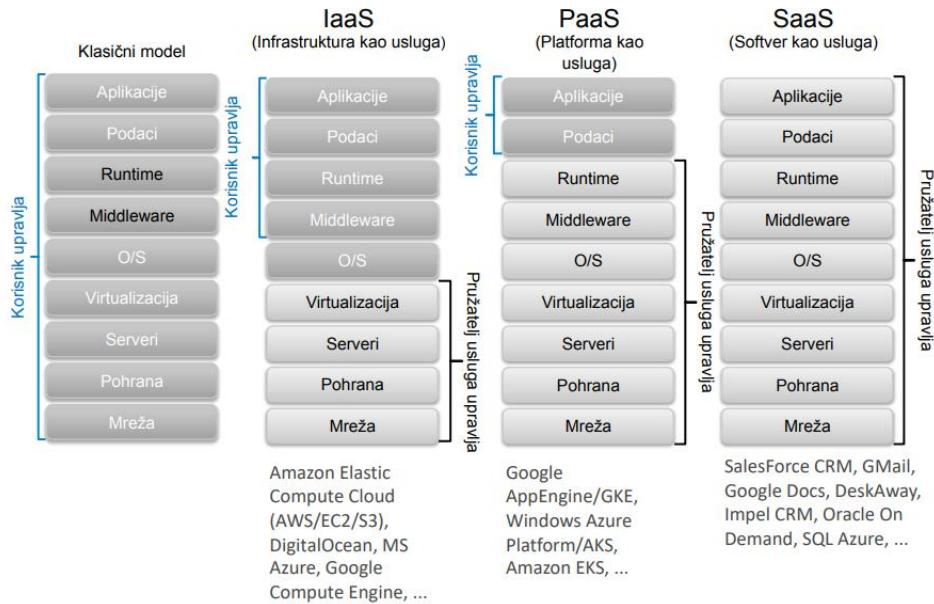
Apstrakcija implementacije

- Korisnik i razvijatelj ne znaju specifikaciju sustava na kojem će se aplikacije i usluge izvoditi
- Podaci spremljeni na lokacije koje nisu poznate
- Administracija sustava nije pod kontrolom razvijatelja
- Pristup aplikacijama i uslugama je omogućen putem Interneta

Virtualizacija

- Omogućuje izvršavanje više operacijskih sustava na jednom fizičkom ili na više fizičkih računala
- Isto vrijedi za pohranu podataka
- Resursi se mogu dijeliti ili udruživati.
- Računala mogu imati podršku za virtualizaciju (tehnologija hipervizora)
- Primjeri: Xen, VMware, Wine, ...

Objasnite modele računarstva u oblaku.



Koja je razlika između virtualnih strojeva i kontejnera? Usporedite virtualne strojeve i kontejnere.

Kontejner je skup izoliranih procesa u korisničkom prostoru OS-a, koristi manje resursa i brže se pokreće od virtualnih strojeva koji moraju pokretati čitav OS. Zbog toga je u kontejneru teže kontrolirati korištenje resursa, izolacija između kontejnera je slaba i svaki kontejner mora koristiti kernel domaćinovog OS-a.



Što je to OCI specifikacija i što definira?

Open Container Initiative specification osnovana je 2015. godine na inicijativu Dockera i ostalih kompanija kako bi definirale specifikacije koje bi svaki kontejner trebao imati (lakša interoperabilnost između različitih alata koji rade s kontejnerima).

Definira:

- Izvršnu specifikaciju
- Specifikaciju slike
- Specifikaciju distribucije slike

Objasnite kako se Docker izvršava na različitim operacijskim sustavima (Linux, Windows, MacOS).

Docker se nativno izvršava na Linuxu, samo ga je potrebno dohvatiti putem terminala. Na Windows-u i MacOS-u potreban je virtualni stroj s Linuxom. Najpoznatije rješenje je Docker Desktop koji na MacOS-u koristi native Hypervisor.framework a na Windows-ima Microsoft Hyper-V kako bi se Docker mogao pokrenuti. Također postoji Docker Machine koji koristi VirtualBox za stvaranje stroja ali se danas rijetko koristi.

Što je to Kubernetes?

Sustav za orkestraciju kotejnera nastao u Google-u 2014. godine. Neovisan je o oblaku u kojem se izvršava, preko pružatelja usluga dostupan je kao PaaS ili IaaS. Korisnik definira krajnje stanje a k8s se brine za održavanje tog stanja.

Koji su osnovni koncepti koji se koriste u Kubernetesu?

node (čvor) – jedno virtualno računalo na kojem se izvršava k8s

pod (kapsula) – najmanji koncept koji se isporučuje, u sebi može imati više kontejnera

replica set – pravila za repliciranje, pazi da određeni broj replika imamo u sustavu

deployment – pravila za čitavu aplikaciju, brine se za verzije slika

service – kada se izvana pristupi usluzi onda se promet preusmjeri na neki od kapsula i na određena vrata (tj. kontejner unutar te kapsule)

namespace – odvaja dijelove k8s grozda može se definirati ograničenje korištenja resursa (quota), mogu se definirati prava korisnika i obično devops tim definira namespace i daje nam prava

label – oznaka nekog koncepta

selector – definira na koje koncepte se odnosi pojedini dio konfiguracije. Odabire se na temelju oznaka (label)

Što je i čemu služi Istio?

Služi za automatsko upravljanje opterećenjem za razne protokole (HTTP, gRPC, WebSocket i TCP). Omogućuje finu kontrolu prometa (pravila usmjeravanja, ponovno slanje zahtjeva,...), definiranje politika (pravo pristupa, ograničenja), automatske metrika, log-ova i trace-ova te omogućuje sigurnu komunikaciju između usluga sa zaštitom identiteta i kriptiranjem komunikacije.

Koji su elementi životnog ciklusa raspodijeljenog sustava?

Definicija zahtjeva, analiza rješenja, sinteza, ispitivanje, rad, mjerenja i modifikacija zahtjeva.

Koji su najvažniji nefunkcijski zahtjevi?

Performanse, raspoloživost i ukupna cijena vlasništva.

Kako se oni skupno zovu?

Kvaliteta usluge (QoS).

Što je to ugovor o razini usluge (SLA)?

Ugovor između korisnika i davatelja usluge koji definira razinu usluge.

Zašto je vrednovanje performansi važno?

Zbog planiranja kapaciteta koji je potreban za uspješan rad

Koje metode analize se koriste u praksi?

Iskustvo, modeliranje i simulacija

Kakve vrste modela tereta postoje u praksi?

Prirodne aplikacije, umjetne aplikacije (benchmarks) i neizvodivi modeli opisani intenzitetom zahtjeva, prosječnim vremenom obrade i sl.

Koji su koraci pri razvoju modela sustava?

Razumijevanje funkcioniranja, modeliranje tereta, mjerenje sustava u pogonu radi utvrđivanja parametara tereta, razvoj modela, verifikacija i validacija, analiza mogućih scenarija promjena, prognoza promjena tereta u budućnosti, prognoza performansi sustava nakon puštanja u pogon te u budućnosti

Koje su najčešće greške kod modeliranja?

Presložena analiza, nema specifičnog cilja, prejudiciranje, nedovoljno razumijevanje sustava, neadekvatne mjere nereprezentativni teret, neuključivanje važnih parametara, promatranje u

krivom intervalu vrijednosti parametara, krivo baratanje ekstremima, nedovoljno promatranje evolucije sustava i tereta, kriva interpretacija rezultata.

Što definiraju pojmovi MTBF i MTTR?

Srednje vrijeme između pogrešaka i srednje vrijeme popravka.

Kako je definirana raspoloživost sustava?

$$D = \text{MTBF}/(\text{MTBF} + \text{MTTR})$$

Postotak vremena u kojem je sustav na raspolaganju korisnicima.

Kako se računa raspoloživost paralelno (D_p) i serijskih (D_s) povezanih sustava?

$$D_p = (1-D_1)*D_2 + (1-D_2)*D_1 + D_1 D_2$$

$$D_s = D_1 * D_2$$

Koje su dvije osnovne grupe troškova za gradnju i pogon web-sustava i kako su obično raspodijeljeni?

Kapitalni i operativni troškovi, grubo raspodijeljeni 50/50 kroz tri godine.

Objasnite pojam vremena odziva i kapaciteta

Vrijeme odziva određuje odziv na jedan zahtjev.

Kapacitet odgovara maksimalnom broju zahtjeva koji se mogu obraditi u jedinici vremena.

Koje se metode za poboljšanje performansi koriste u arhitekturi raspodijeljenih sustava?

Serijsko preklapanje, paralelno preklapanje, paralelno izvođenje i privremena pohrana.

Kako je definirano ubrzanje?

Ubrzanje je omjer vremena izvođenja na jednom i više paralelnih podsustava.

Što su tipični uzroci natjecanja za sredstva (contention) ili potrebe za usklađivanjem podataka (coherence)?

- Zajedničke funkcije i varijable u operacijskom sustavu
- Izmjena zajedničkih podataka koji se mijenjaju u privremenim spremnicima (cache)
- Promet podataka u/iz glavne memorije
- Čekanje na ulaz/izlaz
- Sinkronizacijski primitivi

Koliko razina protokola uključuju web-aplikacije kojima se pristupa putem lokalnih mreža Ethernet spojenih na Internet?

Četiri: Ethernet, IP, TCP i HTTP

Što uključuje vrijeme odziva web-aplikacije?

Vrijeme pristupa klijenta (klijent – ISP), vrijeme prijenosa paketa od klijentovog do poslužiteljevog ISP-a (ISP – ISP), vrijeme dostupa do poslužitelja (ISP – poslužitelj), ISP – davatelj internetske usluge (Internet Service Provider).

Kako se može ostvariti razmjeran rast aplikacije?

Vertikalno ili horizontalno

Koje probleme donosi horizontalan rast?

Usklađivanje, raspodjela tereta i raspodjela podataka

9. Mikrousluge (mikroservisi), komunikacija temeljena na događajima

Koji su nedostaci monolitne arhitekture u odnosu na mikrousluge?

Monolitna arhitektura vezana je za jednu tehnologiju, tj. vezana je za jedan programski jezik, programske okvire i obično relacijske baze podataka. Nije moguće skalirati samo dio aplikacije već cijelu aplikaciju u također je potrebno shvaćanje rada cijele aplikacije. Svaka pojedinačna usluga mikrousluge može se zasebno isporučiti dok se kod monolitne arhitekture mora isporučiti čitava aplikacija.

Na koji način se radi komponentizacija monolita na mikrousluge?

Svaka usluga mora činiti jednu poslovnu funkcionalnost (Single Responsibility Principle). Potrebno je pažljivo definirati sučelja, ako se promijeni jedna mikrousluga ne smije se zahtijevati promjena drugih mikrousluga. Ako se su sučelja mijenjaju treba ih napraviti tako da promjene ne uzrokuju trenutnu promjenu kod drugih usluga.

Navedite 4 problema s kojima se suočavamo kada implementiramo mikrousluge.

Kako pratiti kompleksnost sustava?

Kako znati koji build, tj. koja verzija je na kojem čvoru?

Kako doći do dnevničkih zapisa (raspodijeljeni logging)?

Kako pratiti obradu jednog zahtjeva (traceability)?

Kako se rješava konfiguriranje velikog broja mikrousluga?

Konfiguracija se stavlja u neku datoteku kako bi se lakše mogla instalirati u drugoj okolini. Može se nalaziti u arhivi u kojoj se isporučuje, konfiguracijska datoteka, varijablama okoline ili ovisi o specifičnom rješenju korištenog oblaka.

Čemu služi registracija mikrousluga u registru?

Omogućuje otkrivanje usluga s poslužiteljske strane (komponenta za uravnoteženje brine o otkrivanju usluga) ili otkrivanje usluge s klijentske strane (klijent brine gdje se nalazi usluga).

Kako možemo raditi uravnoteženje opterećenja u sustavu s mikrouslugama?

Ako koristimo Kubernetes onda postoji posebna komponenta u grozdu koja radi uravnoteženje ravnomjerno s obzirom na broj instanci. Istio koristi proxy kod svake usluge kako bi puno finije mogao usmjeravati promet ovisno o opterećenju.

Čemu služi API Gateway?

Jedinstvena točka pristupa za klijente, prihvata HTTP(s)/WebSocket zahtjeve i preusmjerava ih na jednu ili više mikrousluga. Moguće je i napraviti API Gateway za svaku vrstu klijenta (mobilni, web, javni pristup,...).

Navedite nekoliko metrika mikrousluga i tehnologije koje služe za njihovo praćenje.

Metrike mikrousluga: zahtjev/sekundi, memorija, procesor, vrijeme odgovora, probe (liveness, readiness).

Tehnologije za njihovo praćenje: Prometheus, Grafana.

Čemu služi agregiranje dnevničkih zapisa (log) i koje tehnologije koristimo za to?

Služi prikupljanju zapisa svih usluga na jedno mjesto, najčešće se koristi ELK stack:

- Elasticsearch – spremanje podataka, indeksiranje i pretraživanje
- Logstash – čita podatke iz datoteke/mreže, transformira ih i spremi u elasticsearch
- Kibana – vizualizacija podataka (grafovi, pite, ...) iz elasticsearcha

Koji problem rješava praćenje (distributed tracing) zahtjeva u sustavi s mikrouslugama?

Olakšava pronalaženja izvora greške ili kašnjenja unutar mikrousluge.

Objasnite način rada arhitektturnog obrasca osigurača.

Osigurava da se izolira dio koji ne radi i da se prestanu slati zahtjevi na taj dio, tako sprječava kaskadni ispad. Za tu funkcionalnost koristi sljedeća stanja:

- **Close** – svi pozivi se prosljeđuju na krajnju uslugu i vraćaju se odgovori kakvi se dobiju od krajnje usluge (uspješno, neuspješno, timeout). Računa se uspješnost poziva, ako padne ispod definirane granice prelazi se u stanje Open.
- **Open** – niti jedan poziv se ne prosljeđuje usluzi, na zahtjev se vraća neka podrazumijevana vrijednost. Postoji definirano vrijeme koliko se usluga nalazi u tom stanju a nakon tog vremena prelazi u stanje Half Open.
- **Half Open** – dio poziva se prosljeđuje usluzi, računa se uspješnost i ako je iznad definirane granice prelazi u stanje Close u suprotnom se vraća u stanje Open.

Navedite i objasnite strategije uvođenja novih verzija usluga.

Recreate

- ugasi staru verziju i pokreni novu

Ramped

- kada imamo više replika onda dodamo jednu novu verziju, pa ugasimo staru, i tako dok sve ne zamijenimo (Kubernetes)

Blue/Green

- instaliramo novu verziju (*blue*) i testiramo da radi, nakon toga samo sav promet preusmjerimo na novu verziju (*switch*)

Canary

- slično Blue/Green samo postepeno postotak sa stare verzije prebacujemo na novu (Istio)

A/B Testing

- samo neke korisnike prebacimo na novu verziju, primjeri kriterija: beta testeri, geografsko područje, korisnici pojedinog preglednika, jezik, OS, veličina ekrana, ... (Facebook, Istio)

Shadow (Dark Launch)

- instaliramo obje verzije i zahtjeve šaljemo na obje, ali samo od stare se odgovori vraćaju korisniku (testiranje u produkciji)

Navedite i objasnite verzioniranje poruka i njihovu kompatibilnost.

Prije slanja i primanja provjerava poruka se provjerava u registru kompatibilnosti sheme, postoje dvije vrste kompatibilnosti:

- **Prema natrag** – primatelj s novom verzijom može primiti stare verzije od pošiljatelja
- **Prema naprijed** – primatelj sa starom verzijom može primati nove verzije

Koji je problem raspodijeljenih transakcija i zašto se ne koriste kod mikrousluga?

?

Objasnite pojam eventualne konzistentnosti.

Sustav će nakon nekog vremena nakon zadnjeg zahtjeva doći u konzistentno stanje.

Objasnite arhitekturni obrazac CQRS (Command Query Responsibility Segregation).

?

Čemu služi raspodijeljena saga?

Kako bi se dugački zahtjevi mogli prekinuti i po potrebi promjene koje su se dogodile poništiti. Mora omogućiti isporuku zahtjeva najviše jednom a kompenzirajući zahtjev se mora isporučiti najmanje jednom (moraju biti idempotentni moraju se moći izvršiti).

Koja su svojstva računarstva na rubu?

Smanjuje se količina podataka koji se prenose mrežom jer se ne moraju svi podaci slati u oblak, povećana sigurnost podataka jer podaci ne izlaze iz lokalne mreže (GDPR), brže vrijeme odziva jer je manje kašnjenje u mreži.

10. Replikacija podataka

Objasnite vezu između replikacije i konzistentnosti.

Raspodijeljeni sustav je konzistentan u nekom vremenskom trenutku ukoliko su sve replike u njemu nalaze u istom stanju.

Objasnite način replikacije u HDFS-u.

- Svaki **DataNode** periodički šalje poruke **NameNode-u**
 - Porukama **HeartBeat** daje do znanja da je i dalje raspoloživ
 - Porukama **BlockReport** dojavljuje koji blokovi podataka su pohranjeni kod njega
- Na osnovu ovih poruka **NameNode**
 - Inicira dodatnu replikaciju ukoliko neki **DataNode**
 - Postane nedostupan (npr. zbog tehničkog kvara) ili
 - Izgubi blokove podataka pohranjene kod njega (npr. u slučaju problema na nekom od njegovih tvrdih diskova)
 - Inicira brisanje viška replika prilikom vraćanja u sustav privremeno nedostupnog **DataNode-a**
 - Prosljeđuje klijentske zahtjeve za pisanje i čitanje blokova podataka samo na ispravne čvorove **DataNode**

Navedite vrste replika i objasnite glavne razlike između njih.

Trajne replike – početni skup replika postavljen na skupu računala povezanih lokalnom mrežom, statička organizacija, većina zahtjeva je čitanje i raspoređivanje zahtjeva na dostupne replike.

Korisničke replike – dohvaćeni podaci spremaju se u lokalne spremnike i potrebno je održavati njihovu konzistentnost pomoću poslužitelja s kojeg su podaci dohvaćeni.

Poslužiteljske replike – sadrži trajne replike koje su dostupne korisnicima, poslužitelj prati vlastito stanje opterećenja, odabire i raspoređuje replike dinamički tijekom rada sustava.

U sustavu replika koji se sastoji od glavnog poslužitelja i n=6 podjednako opterećenih pomoćnih poslužitelja, izračunajte prosječno mrežno opterećenje glavnog poslužitelja za sljedeće metode održavanja konzistentnosti: a) pull, b) push s prosljeđivanjem novog sadržaja c) push s prosljeđivanjem operacija za promjenu sadržaja i d) push s prosljeđivanjem obavijesti o promjeni sadržaja. Pri tome pretpostavite da korisnike poslužuju samo pomoćni poslužitelji, da je prosječna frekvencija upita $f_u=500$ upita/s, prosječna frekvencija promjena $f_p=4$ promjene/min te da su prosječne veličine replika, upita/odgovora i operacija za promjenu sadržaja replika $l_r=800$ kb, $l_p=10$ kb i $l_o=600$ kb.

?

Koja je razlika između aktivne i pasivne replikacije? Koje jedinice grupne komunikacije se koriste u jednoj i drugoj?

U aktivnoj replikaciji klijent komunicira s grupom replika koristeći TOCAST, zahtijeva isporuku poruka u istom redoslijedu svim replikama u skupini. U pasivnoj replikaciji klijent komunicira s glavnim replikom koristeći VSCAST, ako se glavna replika raspadne mora se izbrati nova glavna replika, zahtijeva da su poruke o promjenama ispravno uređene čak i nakon ispada glavne replike.

Koji je glavni problem aktivne replikacije i kako ga rješava polu-aktivna replikacija?

Glavni problem aktivne replikacije je determinizam provođenja operacija kod pojedine replike (redoslijed njihovog provođenja drugačiji je od redoslijeda prispjeća). Polu-aktivna replikacija rješava taj problem tako da glavna replikacija donosi odluku i obavještava ostale replike, kao kod pasivne replikacije.

Koja je sličnost između aktivne replikacije kod raspodijeljenih sustava i revne replikacije s promjenama bilo koje replike kod raspodijeljenih baza podataka?

?

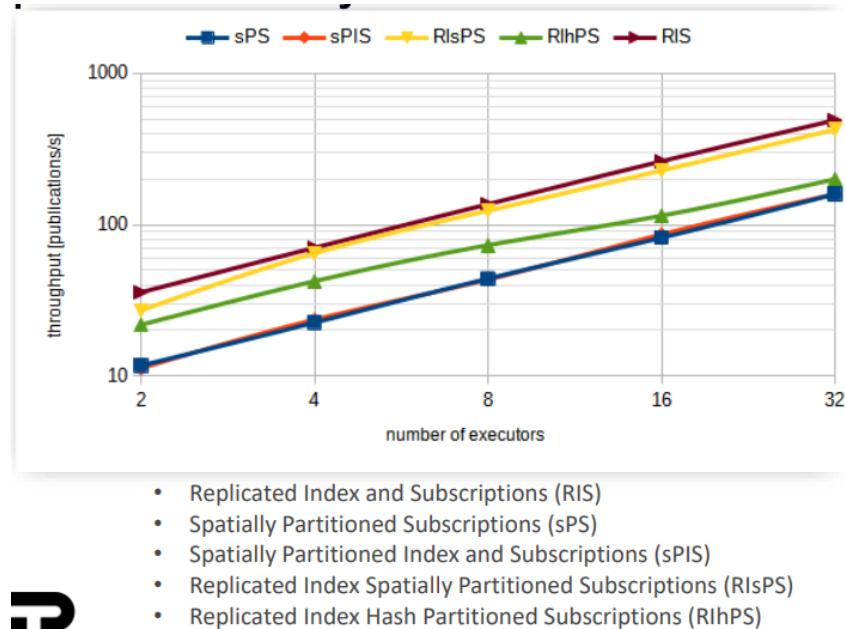
Zašto je neophodno usuglašavanje kod lijene replikacije s promjenama bilo koje replike?

Koordinacija u slučaju s promjenama bilo koje replike (update everywhere) je problematična zbog mogućih konflikata u transakcijama, zato je potrebno usuglašavanje da se odredi koje transakcije trebaju provesti a koje odbaciti.

11. Partitioniranje i konzistentnost podataka

Objasnite postiže li se partitioniranjem uvijek bolja skalabilnost od korištenja replikacije.

Podaci su često indeksirani, a veći indeks najčešće ima bolje performance od partitioniranog indeksa (tj. puno manjih indeksa particija), znači partitioniranje ne more uvijek biti od skalabilnosti.



Objasnite asimetričnost particija na jednom primjeru.

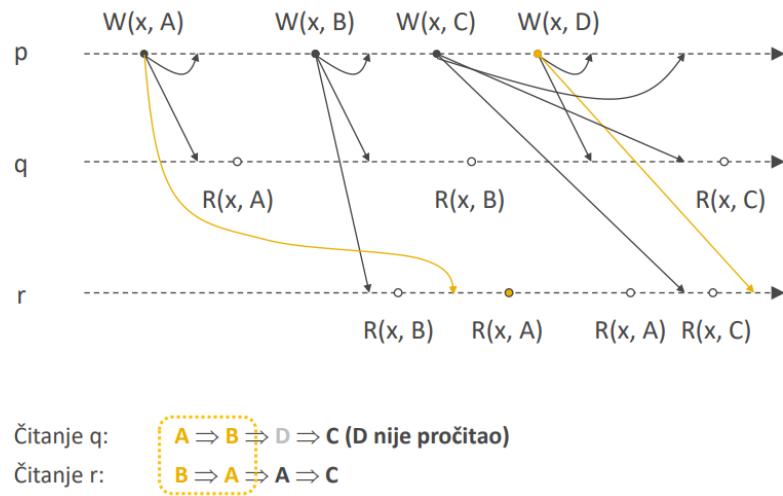
Particije su nejednakih veličina, može se pojaviti tijekom duljeg rada raspodijeljenog sustava. Dovodi do nejednolikog opterećenja čvorova u sustavu i degradira performance čitavog raspodijeljenog sustava.

Koji je nedostatak partitioniranja u krug odnosu na nasumično partitioniranje?

Kod partitioniranja u krug potrebno je repartitioniranje svaki put kada se promijeni broj particija.

Objasnite poštuje li se slijedna konzistentnost u slijedu izvođenja operacija prikazanom na slajdu 35.

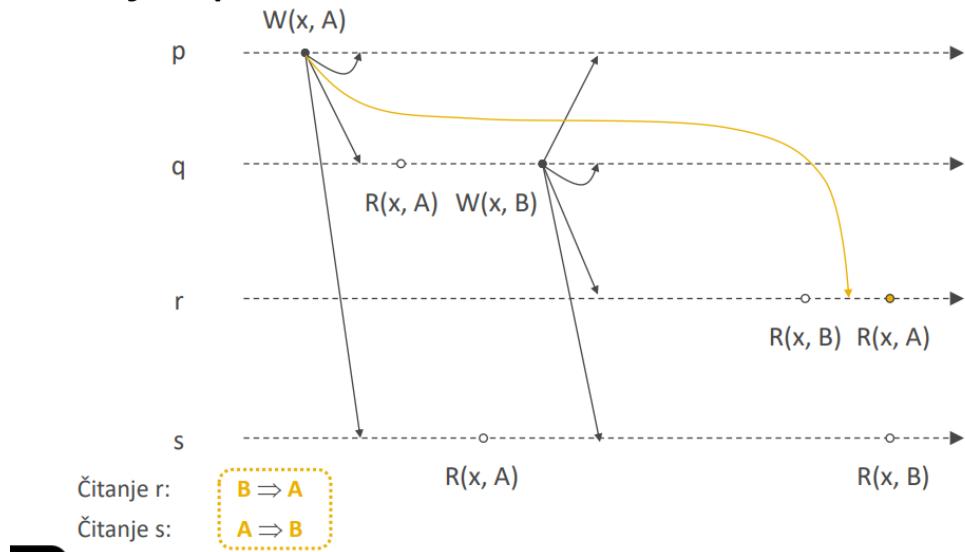
Primjer slijedne nekonzistentnosti



Primjer je za nekonzistentnost tako da ne poštuje se slijedna konzistentnost.

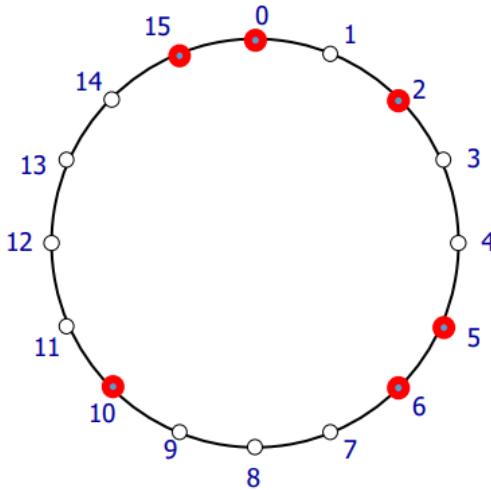
Objasnite poštuje li se povezana konzistentnost u slijedu izvođenja operacija prikazanom na slajdu 38. ukoliko prepostavimo da je operacija pisanja podatka B (uzročno) povezana s operacijom pisanja podatka A.

Primjer povezane nekonzistentnosti



Primjer je za nekonzistentnost tako da ne poštuje se povezana konzistentnost.

12. Sustavi s ravnopravnim sudionicima



Za Chordov prsten sa slike navedite tablicu usmjeravanja za čvor 5.

Prsten ima 16 čvorova, $N = 16$, tako da je $m = 4$. Tablica se zatim računa ovako:

Indeks	Čvor susjed
$5 + 2^0 = 6$	6
$5 + 2^1 = 7$	10
$5 + 2^2 = 9$	10
$5 + 2^3 = 13$	15

Navedite korake koji će pohraniti podatak d za koji vrijedi $H(d)=13$ ako podatak želi pohraniti čvor čiji je ključ jednak 2. Na kome čvoru će podatak biti pohranjen?

Upit kreće iz čvora 2, u svojoj tablici gle koji mu je susjed najbliži čvoru 13 (budući da čvor 13 ne postoji gleda se prvi najbliži a to je čvor 15). Na indeksu $2 + 2^3 = 10$, a čvor 10 na indeksu $10 + 2^2$ ima susjeda čvor 15 na kojeg se zatim posprema podatak d.

Čvor 2 -> Čvor 10 -> Čvor 15

Navedite korake za pronalaženje podatka d iz prethodnog pitanja ako upit za d dolazi s čvora s ključem 0. Koja je cijena u smislu komunikacije među peerovima?

Upit za traženje kreće iz čvora 0, budući da svaki čvor zna na temelju podataka ključ gdje se nalazi podatak čvor 0 zna da mora doći do čvora 15. Na indeksu $0 + 2^3$ pronalazi čvor 10, a čvor 10 na indeksu $10 + 2^2$ ima upisan čvor 15 na kojem se nalazi podataka d.

Čvor 0 -> Čvor 10 -> Čvor 15

13. Tehnologija raspodijeljene glavne knjige

Nema pitanja za ponavljanje

Pitanje 1

Točno

Broj bodova:
1,00 od 1,00

▼ Označi
pitanje

Raspodijeljene sustave (naravno samo one koji su kvalitetno oblikovani i implementirani) karakterizira sljedeće:

- a. inherentna raspodijeljenost, komunikacija razmjenom poruka, uravnoteženo opterećenje✓
- b. raspodjela opterećenja, skalabilnost, potpuna vremenska usklađenost procesa
- c. centralizirano praćenje globalnog stanja, paralelna obrada informacija, dijeljenje sredstava
- d. pouzdanost, otvorenost, serijska obrada informacija
- e. otvorenost, transparentnost, skalabilnost✓

Vaš odgovor je točan.

Ispravni odgovori su: otvorenost, transparentnost, skalabilnost,
inherentna raspodijeljenost, komunikacija razmjenom poruka, uravnoteženo opterećenje

Pitanje **2**

Točno

Broj bodova:
1,00 od 1,00

✓ Označi
pitanje

Tranzijentna komunikacija:

- a. je nepouzdana, garantira isporuku poruke samo ako su pošiljatelj i primatelj poruke istovremeno dostupni.
-
- b. blokira pošiljatelja do primitka potvrde o isporuci poruke primatelju
- c. je ona kod koje klijent registrira zahtjev i "sluša" odgovor, poslužitelj šalje odgovor nakon što završi obradu zahtjeva
- d. omogućuje pošiljatelju nastavak procesiranja odmah nakon slanja poruke
- e. garantira isporuku poruke, poruka se pohranjuje u sustavu i isporučuje primatelju kada je to moguće

Vaš odgovor je točan.

Ispravan odgovor je:

je nepouzdana, garantira isporuku poruke samo ako su pošiljatelj i primatelj poruke istovremeno dostupni.

Pitanje 3

Točno

Broj bodova:
1,00 od 1,00

✓ Označi
pitanje

Ako neki primjer izvođenja operacija pisanja i čitanja raspodijeljenih procesa u vremenu zadovoljava **model povezane konzistentnosti**, koji od sljedećih modela konzistentnosti taj isti primjer zadovoljava?

a. konzistentnost redoslijeda upisivanja

Redoslijed izvođenja operacija pisanja provedenih od strane jednog procesa vidljiv je na jednak način svim ostalim procesima, ali redoslijed izvođenja operacija pisanja različitih procesa može biti vidljiv na proizvoljan način ostalim procesima

b. eventualnu konzistentnost

c. slijednu konzistentnost

Slijed izvođenja operacija može biti proizvoljan, ali svi procesi moraju na jednak način vidjeti konačni slijed izvođenja akcija u vremenu

d. slabu konzistentnost

Ostvaruje se primjenom sinkronizacijskih varijabli koje ostvaruju upravljanje trenucima sinkronizacije replika u raspodijeljenoj okolini

Vaš odgovor je točan.

Ispravan odgovor je:

konzistentnost redoslijeda upisivanja

Pitanje 4

Netočno

Broj bodova:
0,00 od 1,00

Označi
pitanje

Uravnoteženje opterećenja je moguće izvesti u sljedećim primjerima:

- a. primjenom semafora u raspodijeljenoj okolini
- b. horizontalnim skaliranjem web-poslužitelja
 - ✓
- c. vertikalnim skaliranjem web-poslužitelja
 - ✗
- d. u hijerarhijskoj skupini repliciranih procesa

Vaš odgovor nije točan.

Ispravni odgovori su: horizontalnim skaliranjem web-poslužitelja,
u hijerarhijskoj skupini repliciranih procesa

Pitanje 5

Točno

Broj bodova:
1,00 od 1,00

▼ Označi
pitanje

Prikazane su metode na **poslužiteljskoj** strani *socketa TCP*. Upišite ispravan redoslijed izvođenja metoda:

listen()	3	▼	✓
bind()	2	▼	✓
read()	5	▼	✓
socket()	1	▼	✓
close()	7	▼	✓
accept()	4	▼	✓
write()	6	▼	✓

Vaš odgovor je točan.

Ispravan odgovor je:

listen() → 3,

bind() → 2,

read() → 5,

socket() → 1,

close() → 7,

accept() → 4,

write() → 6

Pitanje **6**

Završeno

Broj bodova:
2,00 od 2,00

Označi
pitanje

Objasnite dvije moguće organizacije skupine procesa koje omogućuju otpornost na ispade procesa te navedite za svaku po jednu prednost.

Dvije moguće organizacije skupine procesa koje omogućuju otpornost na ispade su skupina ravnopravnih procesa i hijerarhijska skupina. Kod obje organizacije replicira se jedan proces kako bi u slučaju ispada ostali u skupini mogli "otkriti" ispad. Poruka se šalje svim procesima u skupini, a razlika u organizacijama je ta da kod skupine ravnopravnih procesa svi procesi imaju jednake uloge te je svaki povezan sa svakim, dok je kod hijerarhijske skupine jedan proces koordinator koji je povezan s ostalim procesima "radnicima" koji nisu međusobno povezani osim preko koordinatora. Prednost ravnopravne skupine procesa je u tome što ukoliko dođe do ispada jednog procesa, ostali su i dalje međusobno povezani, no kod hijerarhijske skupine ukoliko dođe do ispada koordinatora, ostali procesi više uopće nisu povezani.

Komentar:

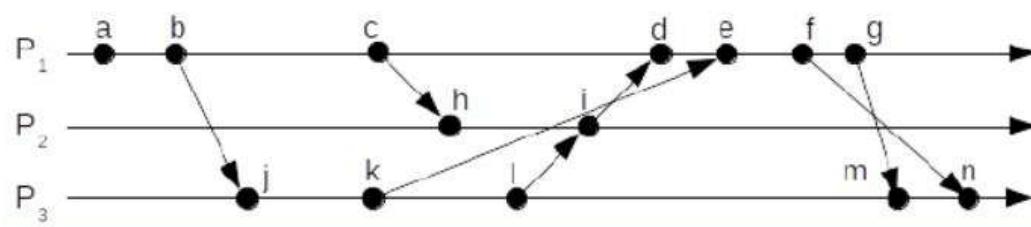


Pitanje 7

Završeno

Broj bodova:
3,00 od 3,00

Označi
pitanje



Na temelju primjera procesa sa slike odgovorite na sljedeća pitanja:

1. Je li komunikacijski kanal od procesa P1 do P2 tipa FIFO (*first in first out*) i zašto?
2. Je li komunikacijski kanal od procesa P1 do P3 tipa FIFO (*first in first out*) i zašto?
3. Zadovoljava li izvođenje događaja na procesima P3 i P1 svojstvo CO (*causal ordering*) i zašto?

1. Komunikacijski kanal od P1 do P2 je FIFO jer je očuvan redoslijed slanja i primanja poruka, tj. poruke koje su prije poslane prije i stižu na odredište.
2. Komunikacijski kanal od P1 do P3 nije FIFO jer poruka bi poruka od **f** trebala stići prije poruke od **g**, no vidimo da je prije stigla poruka od **g** do **m** nego poruka od **f** do **n**.
3. Izvođenje događaja na procesima P3 i P1 ne zadovoljava CO jer bi poruka od **k** do **e** trebala stići prije nego poruka od **l** preko **i** do **d**, no vidimo da to nije slučaj i da je poruka od **l** došla do P1 ranije čime se narušava uzročna slijednost.

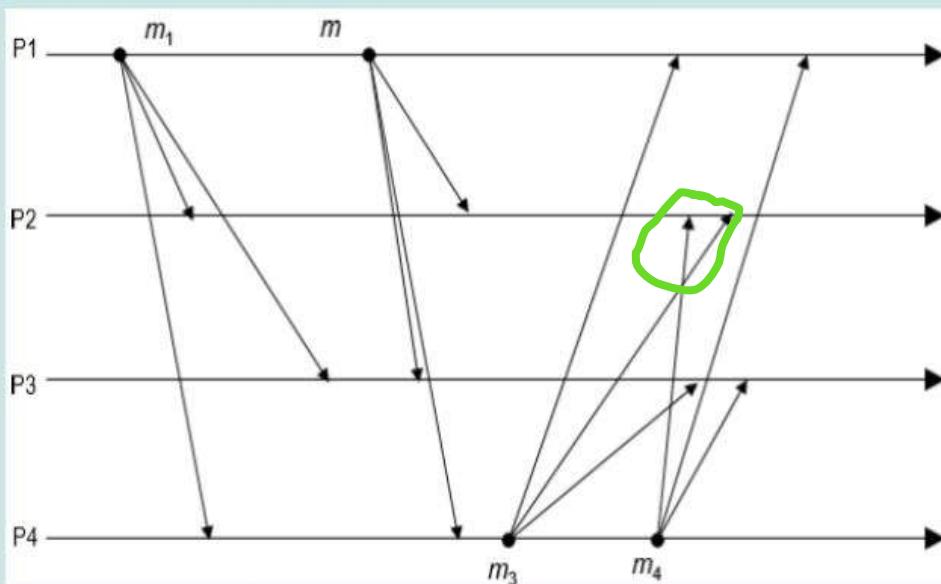
Pitanje 8

Završeno

Broj bodova:
3,00 od 3,00

Označi
pitanje

Objasnite zašto je za pouzdanu komunikaciju skupine procesa od velike važnosti kojim slijedom procesi skupine primaju poruke. Navedite koja je pretpostavka za izvođenje pouzdanog virtualno sinkronog *multicasta* s potpuno uređenim slijedom poruka (tzv. *atomic multicast*). Zadovoljava li sljedeći dijagram svojstva *atomic multicasta*? Objasnite.



Slijed primanja poruka u komunikaciji skupine procesa važan je jer ukoliko svi procesi primaju poruke u jednakom redoslijedu (kojim su poruke poslane), imaju isti pogled na sustav i promjene lokalnih stanja.

Pretpostavka za izvođenje *atomic multicasta* jest upravo da se poruke isporučuju istim redoslijedom svim procesima te da su sve poruke od istog procesa u FIFO redoslijedu.

Dani dijagram ne zadovoljava svojstvo *atomic multicasta* jer se poruka m_3 od procesa P4 isporučuje na procesu P2 nakon poruke m_4 od procesa P4 što narušava pravilan slijed budući da je poruka m_3 poslana ranije s istog procesa kao i m_4 , a obje se isporučuju na istom procesu P2.

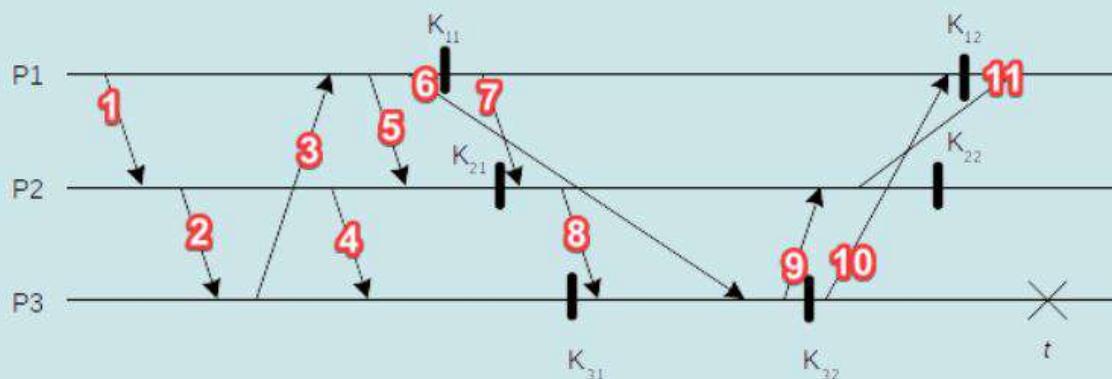
Pitanje **9**

Završeno

Broj bodova:
1,00 od 2,00

Označi
pitanje

Slika prikazuje tri procesa i razmjenu poruka među njima. Svaki proces neovisno o drugim procesima bilježi svoja stanja u označenim kontrolnim točkama. U trenutku t dolazi do ispada procesa P3. Možemo li sustav od tri procesa na slici nakon ispada procesa P3 vratiti u konzistentno stanje koristeći kontrolne točke K12, K22 i K32? Objasnite zašto je to moguće ili nije moguće.



Sustav možemo ne vratiti u konzistentno stanje koristeći K12, K22 i K32 jer su u tim kontrolnim točkama zapamćene sve poslane poruke koje su i primljene (prvih 10 poruka), no poruka 11 je izgubljena jer ju je proces P2 poslao, no njeno je primanje zabilježeno nakon kontrolne točke K12 te zato neće biti zapamćeno.

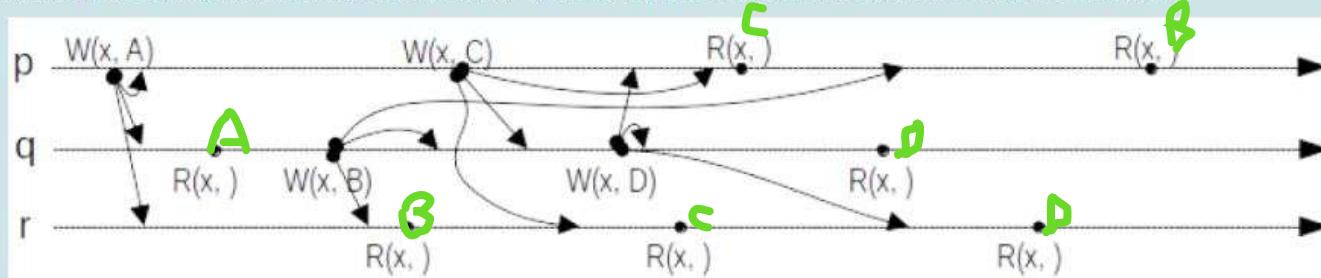
Pitanje 10

Završeno

Broj bodova:
3,00 od 3,00

Označi
pitanje

Na slici je prikazan redoslijed izvođenja operacija triju procesa. Operacija $R(x, d)$ označava čitanje podatka d na lokaciji x . Analogno vrijedi za operaciju $W(x, d)$. Objasnite što je povezana konzistentnost izvođenja operacija. Dovršite prikazani slijed događaja, odnosno popunite operacije čitanja. Poštuje li (ili ne poštuje) prikazani slijed izvođenja operacija povezanu konzistentnost? Objasnite zašto.



Povezana konzistentnost izvođenja operacija znači da je redoslijed izvođenja povezanih operacija **pisanja vidljiv svim procesima** na jednak način, a redoslijed **nepovezanih operacija** pisanja može procesima biti vidljiv na različit način.

p: $R(x, C), R(x, B)$

q: $R(x, A), R(x, D)$

r: $R(x, B), R(x, C), R(x, D)$

Prikazani slijed poštuje povezanu konzistentnost jer su povezane operacije pisanja vidljive svim procesima na jednak način - povezane su operacije pisanja $W(x,A) \rightarrow W(x,B)$, $W(x,C) \rightarrow W(x,D)$ i $W(x,B) \rightarrow W(x,D)$, te se na sva tri procesa podaci A čitaju prije B, C i D, podaci B čitaju se prije podatka D te podaci C prije podatka D, a ostale operacije nisu povezane pa nije važno kojim se redoslijedom čitaju ti podaci.

Dohvaćanje promjena sadržaja (2)

▶ Replace

- Značajke dohvaćanja promjena sadržaja
 - Pogodno za korištenje u slučajevima **čestih izmjena** sadržaja replika tj. kad je frekvencija promjene sadržaja (pisanja) puno veća od frekvencije čitanja
 - Poslužitelji trajnih replika ne moraju znati broj i identitet korisnika
 - Smanjuje se mrežno opterećenje i rastereće poslužitelj replika
 - U slučajevima da lokalno stanje replike nije obnovljeno povećava se vrijeme dohvata novog stanja replika

Pitanje 11

Završeno

Broj bodova:
0,00 od 2,00

☛ Označi
pitanje

Na sljedeća dva pitanja odgovorite s DA/NE i pojasnite zašto je tvrdnja točna ili netočna.

Za metode održavanja konzistentnosti replika *pull* (dohvaćanje promjena sadržaja) i *push* (prosljeđivanje promjena sadržaja) vrijedi sljedeće:

1. metoda *pull* postiže veći stupanj konzistentnosti od metode *push*
2. metoda *push* je pogodna za korištenje u slučajevima rijetkih izmjena sadržaja replika i čestih upita korisnika
 1. DA puno izmjena je dobro za pull
 2. DA

$$2 \text{ paralelna} = R_1(1-R_2) + R_2(1-R_1) + R_1R_2$$

Rserija= $R_1 \times R_2$

MTBF - srednje vrijeme između kvara
 MTTR - mean time between failure
 $MTBF = 3g = 3 \times 8760\text{h}$
 $MTTR = 2\text{h}$
 $3g/(2h+3g) = 0.9999239$
 $MTBF/(MTBF+MTTR)$

Pitanje 12

Točno

Broj bodova:
1,00 od 1,00

Označi
pitanje

Raspodijeljeni sustav uključuje 2 paralelna diska s raspoloživošću od 99.7% te jednu mrežnu sklopku za raspodjelu tereta. Izračunajte raspoloživost sustava ako je srednje vrijeme između kvarova 3 godine, dok je srednje vrijeme popravka 2 sata.

Odgovor: 0,9999



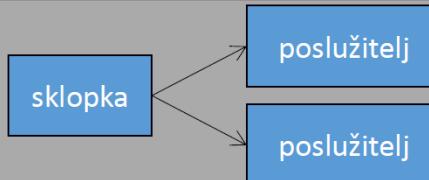
$$0.997(1-0.997)+(1-0.997)0.997+0.997*0.997 =$$

$$0.999991 * 0.9999239$$

$$0.9999149$$

The correct answer is: 0.999914903 +/- 0.01

Primjer: raspoloživost web-poslužitelja



Izračunajte raspoloživost sustava koji uključuje dva paralelna web-poslužitelja s raspoloživošću od 0,99 te jedne mrežne sklopke za raspodjelu tereta između poslužitelja s $MTBF = 1$ godina i $MTTR = 2$ sata.

- Raspoloživost sklopke: $(365 \times 24) / (365 \times 24 + 2) = 0,9998$

- Raspoloživost 2 paralelna poslužitelja:

$$0,0099 + 0,0099 + 0,9801 = 0,9999 \quad 0.99(1-0.99)+0.99(1-0.99)+0.99*0.99$$

- Raspoloživost sustava: $0,9998 \times 0,9999 = 0,9997$

$$100\% / 5\text{ms} = 1 / 0.005 = 200$$

100% je zauzet kroz neko vrijeme

Pitanje **13**

Netočno

Broj bodova:
0,00 od 1,00

▼ Označi
pitanje

Raspodijeljeni sustav uključuje 2 paralelna diska s raspoloživošću od 99.7% te jednu mrežnu sklopku za raspodjelu tereta. **Kolika je maksimalna propusnost pojedinog diska (izražena kao operacije u sekundi) ako U/I operacija diska traje prosječno 5 ms?**

Odgovor: 199.4



The correct answer is: 200 +/- 0

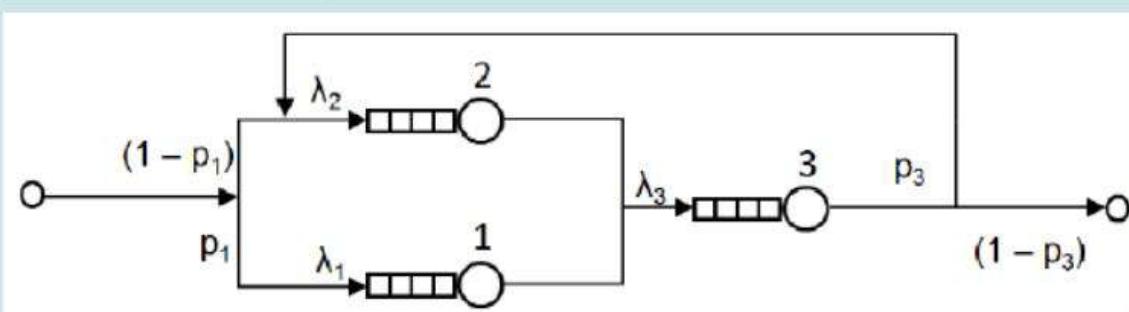
Pitanje 14

Netočno

Broj bodova:
0,00 od 3,00

✓ Označi
pitanje

Zahtjevi dolaze u složeni raspodijeljeni sustav prikazan slikom koji sačinjavaju podsustavi 1, 2 i 3 intenzitetom 0.5 zahtjeva u sekundi. Izračunajte ukupno vrijeme zadržavanja zahtjeva u raspodijeljenom sustavu ako su zadane vjerojatnosti $p_1=0.5$ i $p_3=0.4$ te srednja vremena posluživanja na podsustavima redom 1 s/z, 0.1 s/z i 0.2 s/z.



Odgovor:



The correct answer is: 1.1906 +/- 0.01

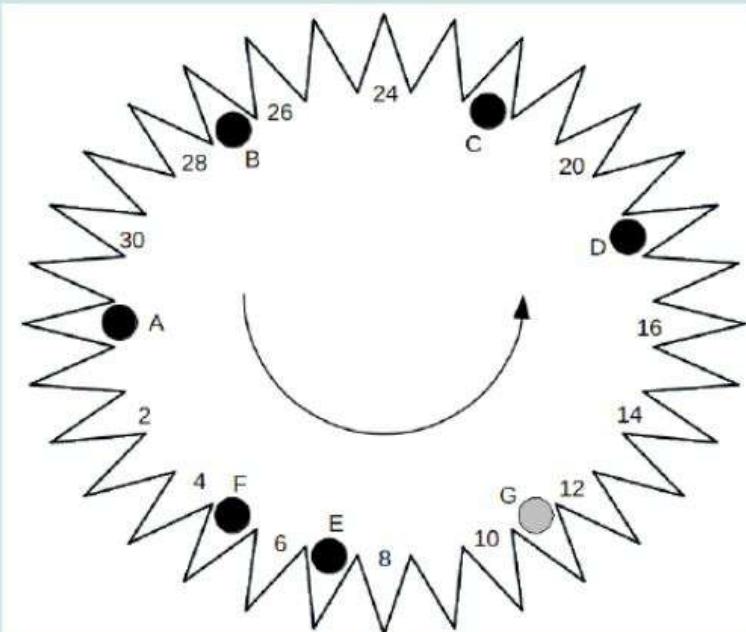
Pitanje 15

Završeno

Broj bodova:
0,00 od 2,00

Označi
pitanje

Na slici je prikazana mreža Chord koja se sastoji od 6 čvorova (A, B, C, D, E, F) i koristi prostor identifikatora duljine $N = 32$ (dovoljno je $m = 5$ bita za kodiranje). Vrijedi sljedeće: $H1(A) = 0$, $H1(B) = 27$, $H1(C) = 22$, $H1(D) = 18$, $H1(E) = 7$ i $H1(F) = 5$. U postojeću mrežu dodaje se novi čvor G za kojega vrijedi $H1(G) = 11$.



Navedite i objasnite 5 koraka koje je potrebno provesti kako bi novi čvor postao funkcionalan čvor u Chordovom prstenu uz pretpostavku da čvor G „poznaće“ čvor A. Napomena: potrebno je navesti korake raspodijeljenog algoritma metode join(11), ali nije potrebno navesti kako izgledaju tablice finger na čvoru G ili ostalim čvorovima prije i nakon dodavanja čvora G u Chordov prsten.

Pitanje **16**

Završeno

Broj bodova:
0,00 od 3,00

Označi
pitanje

Za fragment block-lanca prikazan donjom slikom koji se sastoji od 3 bloka, navedite je li lanac ispravan ili nije te pojasnite zašto je ispravan ili nije ispravan (navesti neispravan blok/blokove, koji dio bloka je neispravan i zašto). Pretpostavka je da algoritam *Proof of Work* zahtijeva hash vrijednost koja započinje s 4 nule.

Block:	#	3
Nonce:	108871	
Data:	Tx4 Tx5	
Prev:	000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd84452cdaf043c19	
Hash:	00007bd4e802a04f15bb2ca5f167fb3c396ceed61c537c94738bf72525eb5c536	
Block:	#	4
Nonce:	133962	
Data:	Tx6 Tx7 Tx8	

<https://andersbrownworth.com/blockchain/blockchain>

Nonce:	133962
Data:	Tx6 Tx7 Tx8
Prev:	00007bd4e802a04f15bb2ca5f167fb3c396ceed61c537c94738bf72525eb6536
Hash:	0000e718ce48d4f121c11077394b0be82d30a81c4f804a97d0f53044d36ea537
Block:	# 5
Nonce:	56265
Data:	Tx9 Tx10
Prev:	0000e718ce48d4f121c11077394b0be82d30a81c4f804a97d0f53044d36ea537
Hash:	bd03ea19218212d987105e42723f849de2c8fe35d8f46a2aa052f66a14ffadf2

kod ovog hasha moraju biti prve 4 nule kako bi znali da nije mijenjan,
odnosno mine-an