

SVEUČILIŠTE U ZAGREBU



Diplomski studij

Informacijska i komunikacijska tehnologija:

Telekomunikacije i informatika

Računarstvo:

Programsko inženjerstvo i informacijski sustavi

Računarska znanost

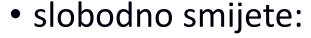
Raspodijeljeni sustavi

9. Replikacija i konzistentnost podataka

Ak. god. 2020./2021.

Creative Commons





- dijeliti umnožavati, distribuirati i javnosti priopćavati djelo
- prerađivati djelo





- imenovanje: morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- nekomercijalno: ovo djelo ne smijete koristiti u komercijalne svrhe.
- dijeli pod istim uvjetima: ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.







U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava. Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava. Tekst licence preuzet je s http://creativecommons.org/



Sadržaj predavanja

- Replikacija i konzistentnost podataka
 - Svrha replikacije podataka
 - Dijeljeni spremnički prostor

Modeli održavanja konzistentnosti podataka

Uspostava replikacije podataka

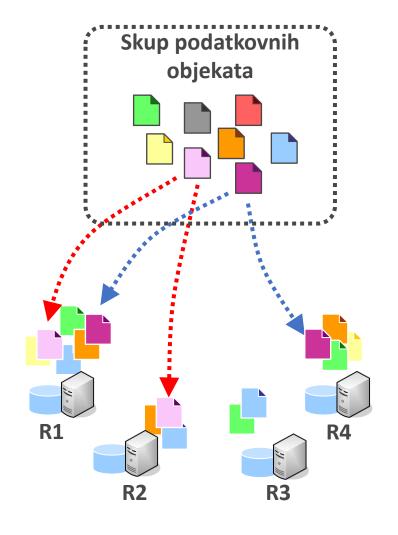


Replikacija

 Raspodijeljeni sustavi koriste podatkovne objekte postavljene na skupu računala

 Podatkovni objekti su spremljeni u obliku nekoliko kopija (replika) na različitim računalima

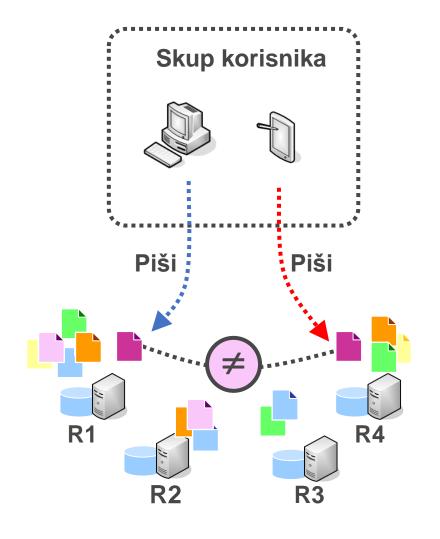
 Replikacija je postupak stvaranja i upravljanja kopijama podatkovnih objekata





Konzistentnost podataka

- Replike istog objekta mogu istodobno i nezavisno koristiti različiti korisnici i mijenjati ih na različite načine u vremenu
- Konzistentnost je narušena kada postoje replike nekog objekta koje nemaju istovjetno stanje
- Nakon nekog vremena postići će se konzistentno stanje – eventualna konzistentnost





Svrha replikacije

Pouzdanost podataka

- U slučaju da neka od replika postane nedostupna, sustav prosljeđuje zahtjeve preostalim dostupnim replikama
- U slučaju da neka od replika ima pogrešne zapise, usporedbom zapisa više replika ostvaruje se otpornost na pogreške u zapisima

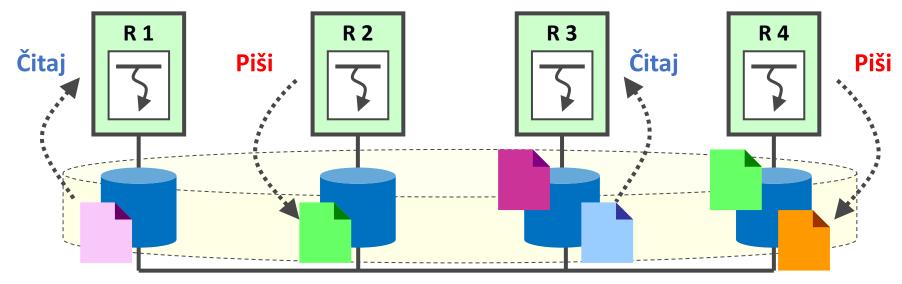
Učinkovito posluživanje podataka

- U slučaju da neka od replika je preopterećena, pristigli zahtjevi prosljeđuju se ostalim replikama
- Replike je moguće predodrediti za posluživanje različitih razreda zahtjeva



Model dijeljenog spremničkog prostora

- Skup spremničkih prostora na računalima u raspodijeljenoj okolini
- Osnovne operacije
 - Čitanje podataka (R)
 - Pisanje podataka (W)





Hadoop Distributed File System (HDFS)

- Primjer dijeljenog spremničkog prostora
- Čini ga grozd računala (skup računala povezanih brzom lokalnom mrežom) koji se sastoji od jednog glavnog čvora NameNode i nekoliko pomoćnih čvorova DataNode
- Svaki podatkovni objekt (datoteka) se u HDFS-u
 - Dijeli na jedan ili više blokova podataka (čija je maksimalna veličina unaprijed predefinirana)
 - Pri čemu je svaki blok podataka (uobičajeno) repliciran 3 puta

Čvorovi DataNode

- Pohranjuju blokove podataka kod sebe
- Obrađuju klijentske zahtjeve za pisanje i čitanje blokova podataka

Čvor NameNode

- Upravlja spremničkim prostorom određuje gdje će biti pohranjene replike pojedinog bloka podataka
- Regulira pristup klijenata pohranjenim podatkovnim objektima jer jedini zna gdje se nalaze replike određenog bloka podataka



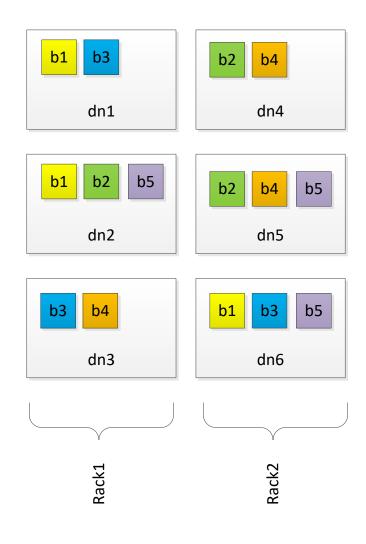
Replikacija u HDFS-u

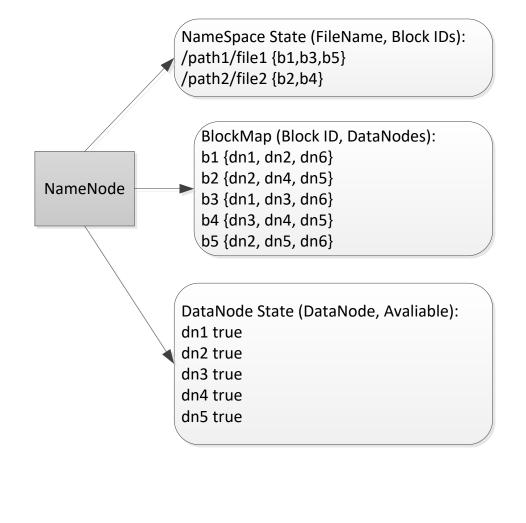
- Svaki DataNode periodički šalje poruke NameNode-u
 - Porukama HeartBeat daje do znanja da je i dalje raspoloživ
 - Porukama BlockReport dojavljuje koji blokovi podataka su pohranjeni kod njega
- Na osnovu ovih poruka NameNode
 - Inicira dodatnu replikaciju ukoliko neki DataNode
 - Postane nedostupan (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



Arhitektura HDFS-a

DataNodes:







Sadržaj predavanja

- Replikacija i konzistentnost podataka
 - Svrha replikacije podataka
 - Dijeljeni spremnički prostor

Modeli održavanja konzistentnosti podataka

Uspostava replikacije podataka



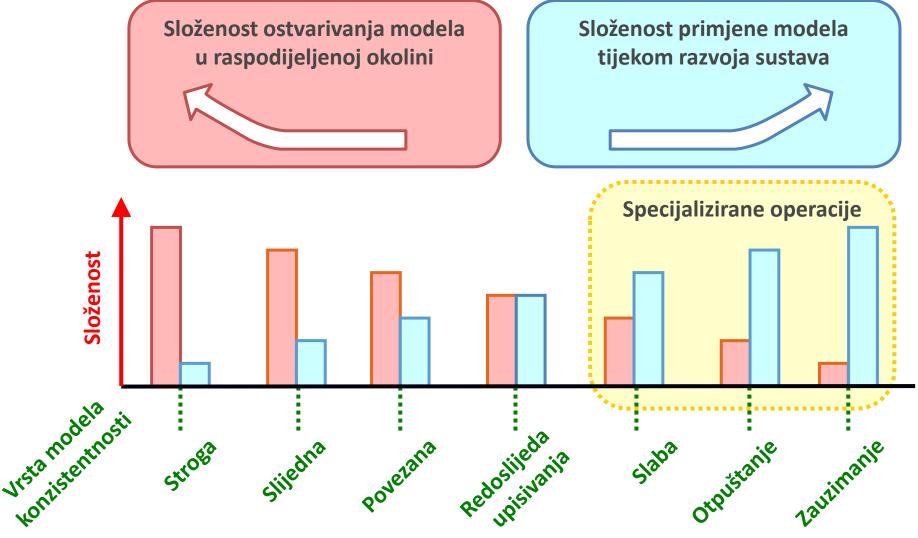
Modeli održavanja konzistentnosti podataka

- Raspodijeljeni sustav podržava određeni model konzistentnosti kada se:
 - podaci u sustavu mijenjaju i te
 - promjene vide samo na načine dozvoljene tim modelom

 Model konzistentnosti stoga uvodi ograničenja na načine mijenjanja podataka u vremenu te na načine na koje se te promjene vide u sustavu



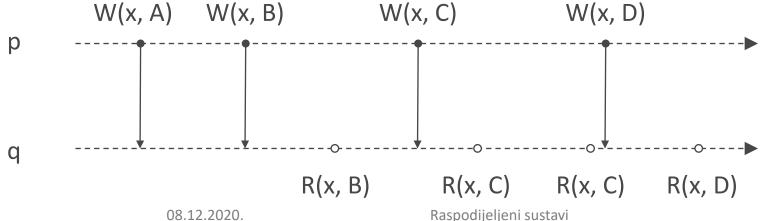
Razredba modela konzistentnosti





Stroga konzistentnost (Strict consistency)

- Čitanje podatka na lokaciji X daje vrijednost koja je posljednja bila zapisana na lokaciju X
 - W(x, Y) zapis podatka Y na lokaciju x
 - R(x, Y) čitanje podatka Y s lokacije x
- Značajke modela
 - Primjena globalnog vremena sinkroni model
 - Ostvaren po uzoru na jednoprocesorske sustave
- Smatran je neizvedivim u praksi (međutim Googleova raspodijeljena baza podataka Cloud Spanner ga podržava jer koristi raspodijeljeni sat TrueTime)



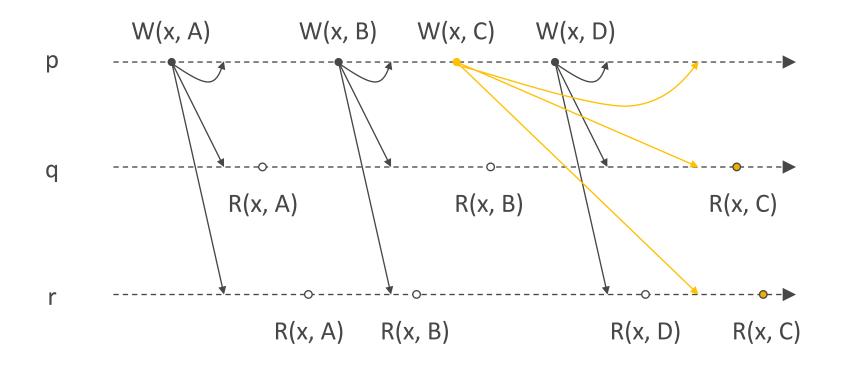


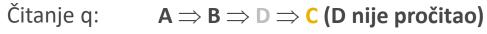
Slijedna konzistentnost (Sequential consistency)

- Slijed izvođenja operacija može biti proizvoljan, ali svi procesi moraju na jednak način vidjeti konačni slijed izvođenja akcija u vremenu.
- Značajke modela
 - Nije potrebno održavati globalni tijek vremena
 - Procesi se moraju dogovoriti o globalnom redoslijedu izvođenja akcija u vremenu
- U praksi se najčešće koristi u raspodijeljenim datotečnim sustavima (DFS-ovima) kao što su DEFS, Calypso, Frangipani, xFS, Microsoft Niobe, Ivy DSM, itd.



Primjer slijedne konzistentnosti

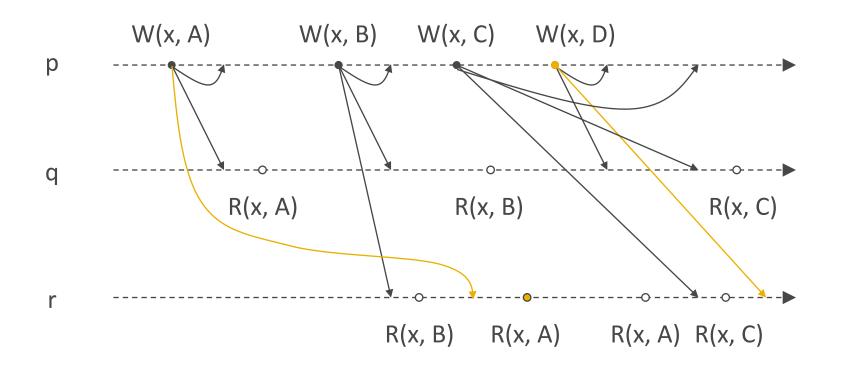




Čitanje r: $\mathbf{A} \Rightarrow \mathbf{B} \Rightarrow \mathbf{D} \Rightarrow \mathbf{C}$



Primjer slijedne nekonzistentnosti





Čitanje q:

 $A \Rightarrow B \Rightarrow D \Rightarrow C$ (D nije pročitao)

Čitanje r: $\mathbf{B} \Rightarrow \mathbf{A} \Rightarrow \mathbf{A} \Rightarrow \mathbf{C}$

08.12.2020.

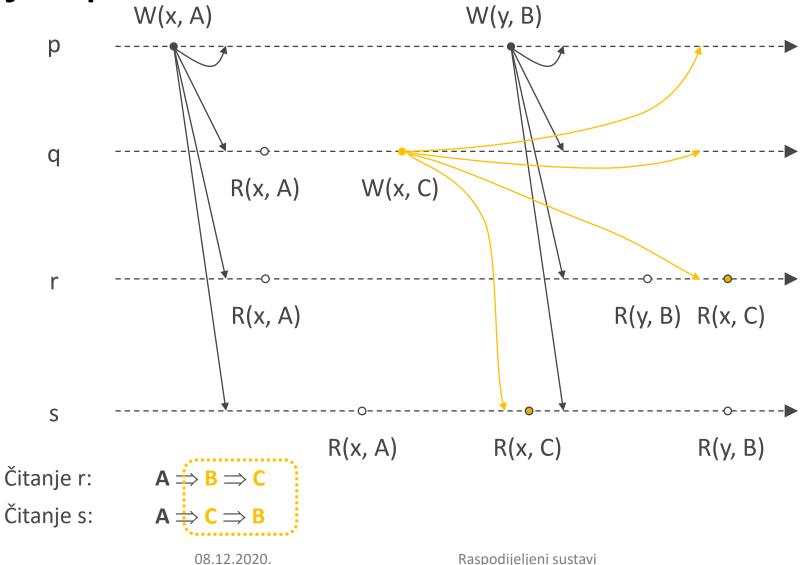
Raspodijeljeni sustavi

Povezana konzistentnost (Casual consistency)

- Redoslijed izvođenja povezanih operacija pisanja vidljiv je svim procesima na jednak način, dok redoslijed izvođenja operacija pisanja koje nisu povezane svakom procesu može biti prikazan na drugačiji način
- Povezanost operacija
 - Operacija pisanja W sadržaja u lokaciju X prethodi operaciji čitanja sadržaja R iz lokacije X čime je operacija R povezana s operacijom W
 - Dvije operacije pisanja nisu povezane ako ostvaruju istodobno zapisivanje sadržaja u različite lokacije dijeljenog spremnika
- U praksi se koristi u nekim NoSQL bazama podataka kao što su MongoDB i AntidoteDB

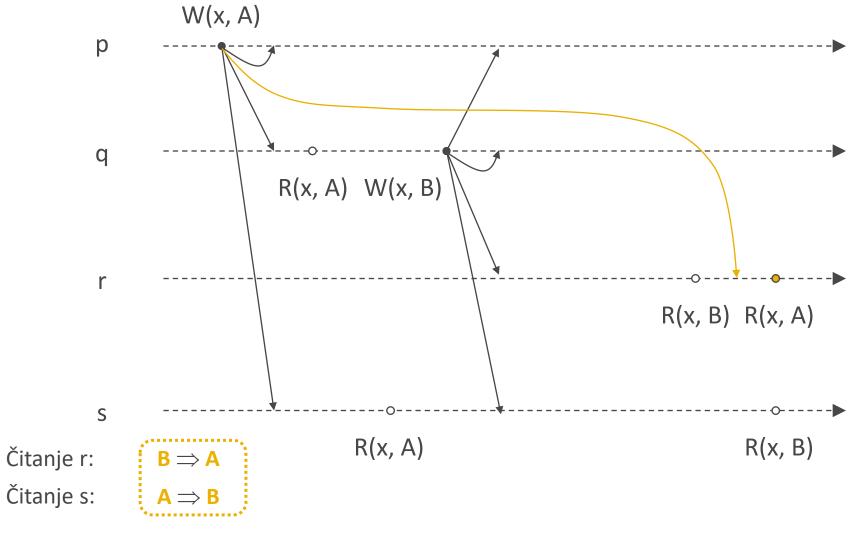


Primjer povezane konzistentnosti





Primjer (ne)povezane nekonzistentnosti



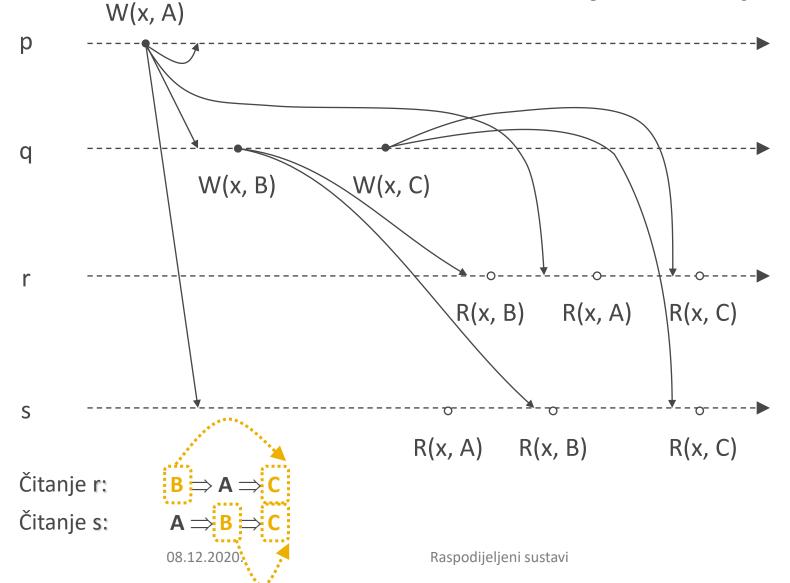


Konzistentnost redoslijeda upisivanja (FIFO con.)

- 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.
- Značajke modela
 - Jednostavno ostvarenje zasnovano na pridruživanju jedinstvenih oznaka svakom zahtjevu za pisanje
 - Jedinstvena oznaka uključuje identifikator procesa i redni broj izvođenja operacije
- U praksi se koristi u nekim raspodijeljenim dijeljenim memorijama



Primjer konzistentnosti redoslijeda upisivanja





Slaba konzistentnost (Weak consistency)

- Ostvaruje se primjenom sinkronizacijskih varijabli koje ostvaruju upravljanje trenucima sinkronizacije replika u raspodijeljenoj okolini
- Operacija sinkronizacije Sync (S)
- Primjena sinkronizacijskih varijabli
 - Usklađivanje svih replika zadanog skupa podataka
 - Svi procesi vide istu vrijednost podataka u trenutku nakon što je provedeno usklađivanje podataka – njihova sinkronizacija
 - Slijed akcija nad sinkronizacijskim varijablama vidljiv je na jednak način svim procesima

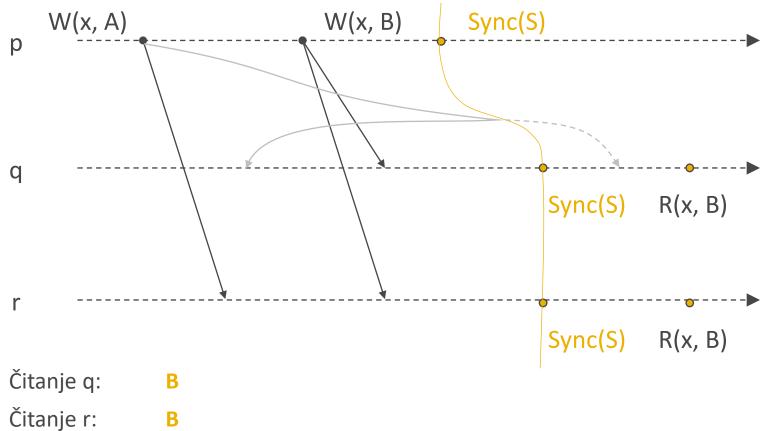


Slaba konzistentnost (Weak consistency)

- Uvjeti primjene i ostvarenja operacije Sync
 - Operacija nad sinkronizacijskom varijablom dovršava se tek nakon što su završene sve prethodno započete operacija pisanja
 - Nove operacije pisanja i čitanja mogu se izvoditi tek nakon završetka izvođenja operacije nad sinkronizacijskom varijablom
- U praksi se koristi u nekim raspodijeljenim strukturama podataka (npr. raspodijeljeno stablo B-Tree)



Primjer slabe konzistentnosti



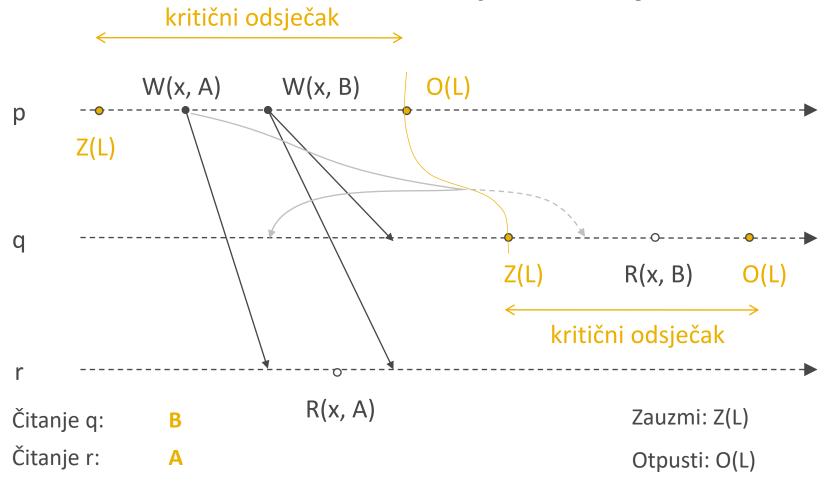


Konzistentnost otpuštanja (Release consistency)

- Vrsta konzistentnosti zasnovana na primjeni kritičnih odsječaka
 - Konzistentnost se održava nakon izlaska iz kritičnog odsječaka
 - Izgradnja kritičnih odsječaka ostvarena je primjenom operacija Zauzmi (Z) i Otpusti (O)
- Operacija Zauzmi
 - Ulazak u kritični odsječak
 - Isključivi pristup spremniku podataka
- Operacija Otpusti
 - Izlazak iz kritičnog odsječka, prije izlaska sve lokalne promjene prosljeđuju se svim replikama podataka
 - Dozvoljen pristup spremniku podataka
- U praksi se koristi u nekim raspodijeljenim dijeljenim memorijama (Munin, Adsmith, itd.), a centralizirani model se koristi u programskim jezicima C++ i Javi



Primjer konzistentnosti otpuštanja



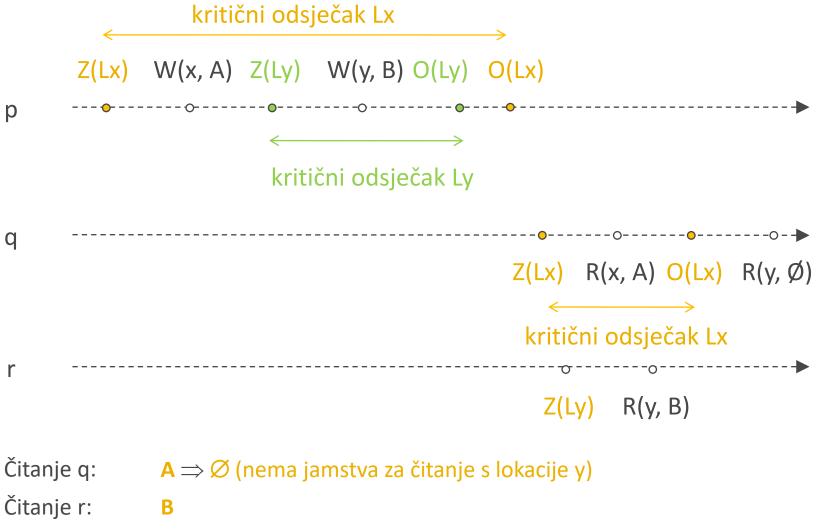


Konzistentnost zauzimanja (Entry consistency)

- Konzistentnost zasnovana primjeni sinkronizacijskih varijabli
 - Konzistentnost se održava u trenutku ulaska u kritični odsječak
 - Izgradnja kritičnih odsječaka ostvarenih primjenom operacija Zauzmi (Z) i Otpusti (O)
- Operacija Zauzmi
 - Ulazak u kritični odsječak, početak izvođenja odsječka tek nakon što su usklađene sve vrijednosti replika s posljednjom promjenom
- Operacija Otpusti
 - Izlazak iz kritičnog odsječaka
- U praksi se koristi u nekim raspodijeljenim dijeljenim memorijama (Midway, DiSOM, itd.)



Primjer konzistentnosti zauzimanja





Sadržaj predavanja

- Replikacija i konzistentnost podataka
 - Svrha replikacije podataka
 - Dijeljeni spremnički prostor

Modeli održavanja konzistentnosti podataka

Uspostava replikacije podataka



Uspostava replikacije podataka

- Uvod u replikaciju podataka
- Organizacija sustava replika
- Razredba vrsta replika
 - Trajne replike, poslužiteljske replike, korisničke replike
- Održavanje konzistentnosti replika
 - Dohvaćanje promjena stanja replika
 - Prosljeđivanje promjena stanja replika
- Ostvarivanje operacija
 - Operacije pisanja sadržaja
 - Operacije čitanja sadržaja



Uvod u replikaciju podataka

- Replika je kopija originalnog podatkovnog objekta (resursa) na nekom drugom računalu u sustavu
- Raspodijeljeni sustav je konzistentan u nekom vremenskom trenutku ukoliko su sve replike u njemu nalaze u istom stanju
- Replikacija je postupak stvaranja i upravljanja replikama
- Dva glavna problema u sustavu replika:
 - Gdje smjestiti replike?
 - Kako održati konzistentnost replika? (Tko će i kada postavljati i ažurirati replike u sustavu?)



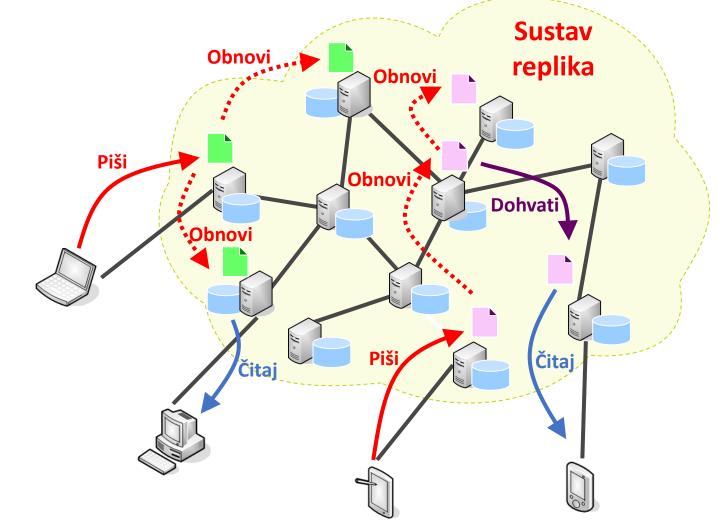
Prednosti i nedostaci replikacije

- Prednosti replikacije
 - Povećanje pouzdanosti sustava (otpornost na kvarove, otpornost na pogreške
 - Povećanje performansi sustava (skalabilnost, odziv)
- Nedostaci replikacije
 - Nekonzistentnost replika
 - Održavanje konzistentnosti replika može generirati veliku količinu mrežnog prometa



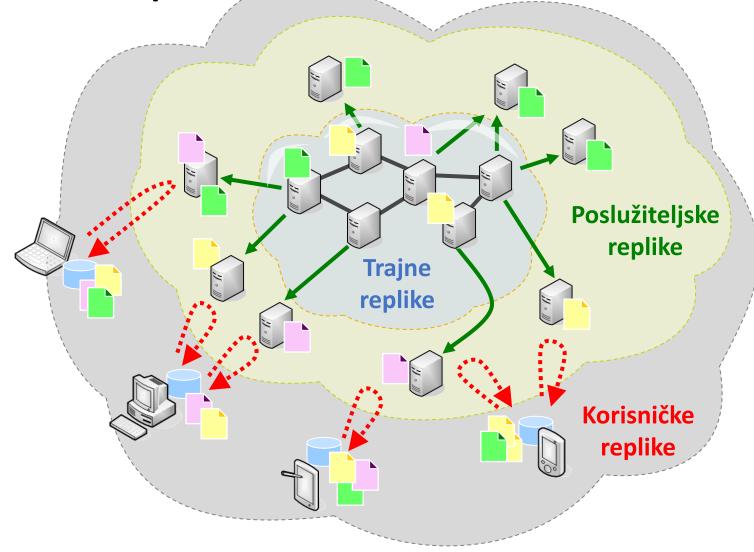
Organizacija sustava replika

- Elementi sustava
 - Računala
 - Spremnici
 - Replike
- Operacije
 - Čitaj
 - Piši
 - Dohvati
 - Obnovi





Razredba vrsti replika



Vrste replika

- Trajne
- Korisničke
- Poslužiteljske



Trajne replike (1)

- Početni skup replika postavljen na skupu računala povezanih lokalnom mrežom
 - Grozdovi poslužitelja
 - Replikacijski poslužitelji

- Osnovne značajke
 - Statička organizacija i postavke sustava
 - Većina zahtjeva je čitanje podataka
 - Raspoređivanje zahtjeva na dostupne replike



Trajne replike (2)

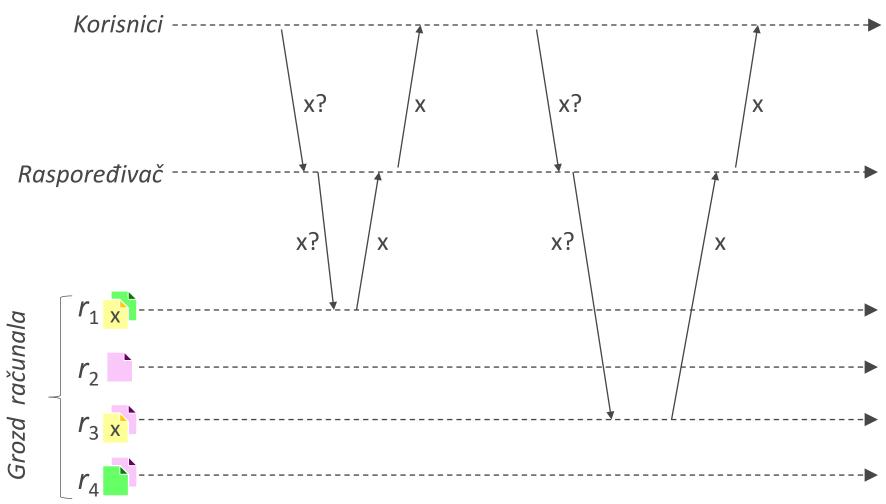
- Organizacija grozda poslužitelja
 - Raspoređivač je pristupna točka sustava koja sadrži informacije koje opisuju poslužitelje unutar grozda
 - Raspoređivač prihvaća i prosljeđuje zahtjeve poslužiteljima
 - Odabir poslužitelja ostvaruje se tako da se ostvari optimalno raspoređivanje opterećenja po replikama koje se nalaze na poslužiteljima
 - Poslužitelji prihvaćaju zahtjev te prosljeđuju rezultate obrade
- Primjer: zastupnik (reverse proxy) poslužitelja weba



37 od 66

Primjer uporabe trajnih replika

x? – zahtjev za dohvat replike tipa X, x – sadržaj replike tipa X





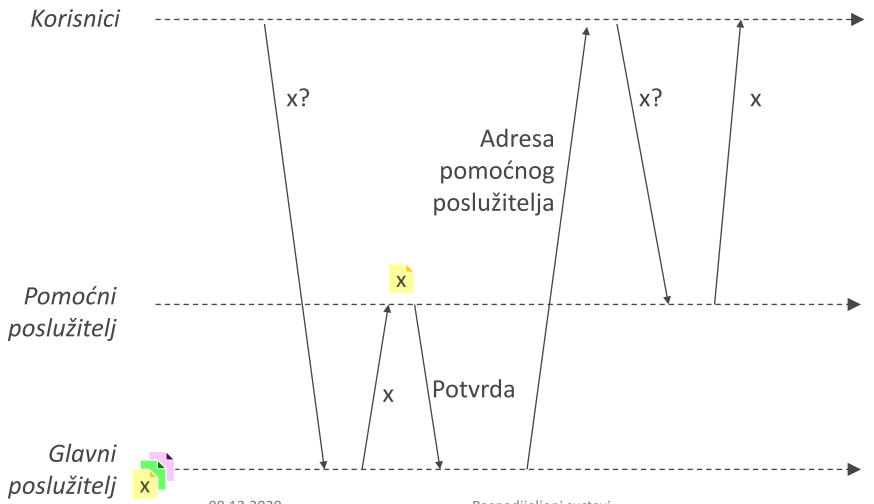
Poslužiteljske replike

- Poslužitelj sadrži trajne replike koje su dostupne korisnicima
 - U slučaju povećanja potražnje podataka, poslužitelj započinje postupak repliciranja podataka
 - Poslužitelj prosljeđuje replike traženih podataka privremenim poslužiteljima
- Osnovne značajke
 - Poslužitelj prati vlastito opterećenje
 - Odabir i raspoređivanje replika ostvaruje se dinamički tijekom rada sustava
- Primjer: regionalna raspodjela opterećenja poslužitelja weba, poslužitelji za VoD (video on demand) kod IPTV-a



Primjer uporabe poslužiteljskih replika

x? – zahtjev za dohvat replike tipa X, x – sadržaj replike tipa X





Korisničke replike (1)

- Korisnički programi koriste lokalni spremnik
 - Dohvaćeni podaci spremaju se u lokalni spremnik
 - U slučaju potrebe za istim podacima, podaci se dohvaćaju iz lokalnog spremnika
 - Potrebno je održavati konzistentnost lokalnog spremnika s poslužiteljem s kojeg su podaci dohvaćeni
 - Lokalni spremnik može biti na istom računalu kao i korisnički programi ili na dijeljenom računalu u lokalnoj mreži
- Osnovne značajke
 - Najpovoljnije je koristiti u slučajevima kada se najčešće provode operacije čitanja
 - Smanjuje se vrijeme dohvata podataka
 - U slučaju kada nekoliko korisnika dijeli lokalni spremnik povećava se učinkovitost primjene korisničkih replika



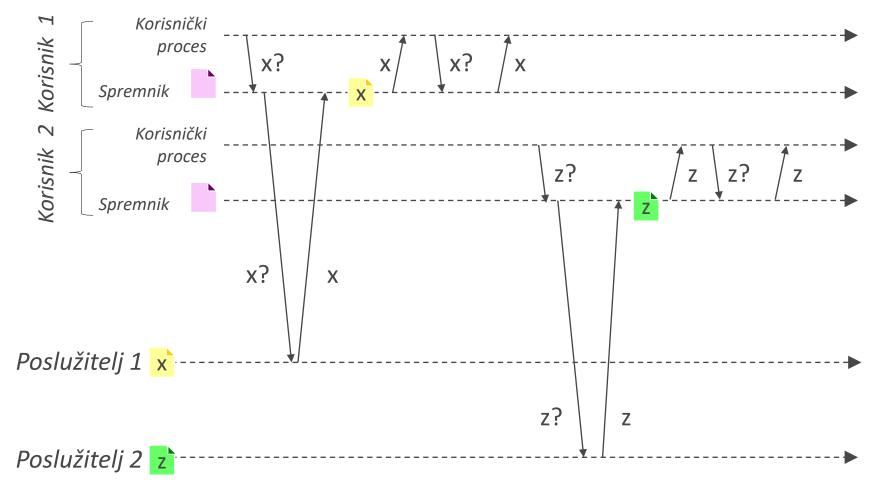
Korisničke replike (2)

- Korisnici ostvaruju replikaciju podataka u lokalnom spremniku
 - Svaki podatak koji korisnik želi dohvatiti s udaljenog poslužitelja prvo traži u lokalnom spremniku
 - Ako se traženi podatak ne nalazi u lokalnom spremniku dokument se dohvaća s udaljenog poslužitelja
 - Ako se podatak već nalazi u lokalnom spremniku, dokument se dohvaća iz lokalnog spremnika
- Primjer: priručno spremište (cache) preglednika weba, forward proxy cache u lokalnoj mreži klijenata



Primjer uporabe korisničkih replika

x? – zahtjev za dohvat replike tipa X, x – sadržaj replike tipa X





Uspostava replikacije podataka

- Uvod u replikaciju podataka
- Organizacija sustava replika
- Razredba vrsta replika
 - Trajne replike, poslužiteljske replike, korisničke replike
- Održavanje konzistentnosti replika
 - Dohvaćanje promjena stanja replika
 - Prosljeđivanje promjena stanja replika
- Ostvarivanje operacija
 - Operacije pisanja sadržaja
 - Operacije čitanja sadržaja



Održavanje konzistentnosti replika

- Obnavljanje stanja replika
 - Korisničke i poslužiteljske replike je potrebno usklađivati s promjenama stanja trajnih replika
 - Obnavljanje sadržaja replika može biti ostvareno u trenutku promjene sadržaja ili u trenutak prije ostvarivanja pristupa replici
- Osnovne metode održavanja konzistentnosti sadržaja replika
 - Dohvaćanje promjena sadržaja (pull)
 - Prosljeđivanje promjena sadržaja (push)



Dohvaćanje promjena sadržaja (1)

 Korisnici dohvaćaju promjene sadržaja trenutak prije pristupa replikama

• Primjenjuju se od strane korisnika Poslužitelji trajnih prema poslužiteljskim replikama i Poslužiteljske replika poslužiteljima trajnih replika replike Korisnici 6



08.12.2020.

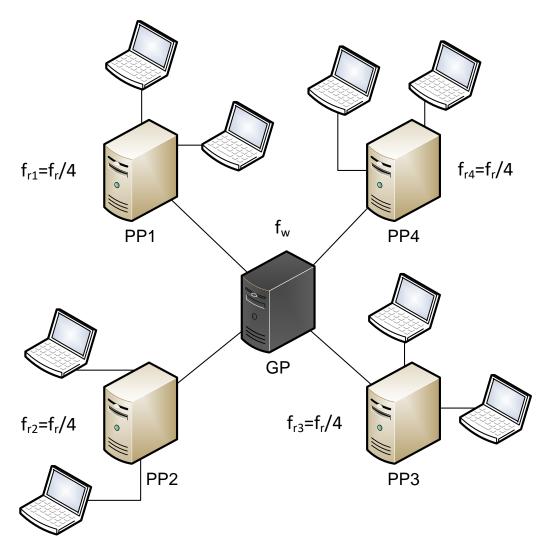
Raspodijeljeni sustavi

Dohvaćanje promjena sadržaja (2)

- 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 rasterećuje poslužitelj replika
 - U slučajevima da lokalno stanje replike nije obnovljeno povećava se vrijeme dohvata novog stanja replika



Primjer [1]



- Klijentski upiti i zahtjevi su ravnomjerno raspoređeni po pomoćnim poslužiteljima
- f_r frekvencija čitanja
- f_w frekvencija pisanja (tj. promjene sadržaja)
- Mrežno opterećenje GP-a?



Primjer [2]

- Parametri
 - f_r frekvencija čitanja
 - f_w frekvencija pisanja (tj. promjene sadržaja)
 - l_m veličina poruke (tj. zahtjeva)
 - l_r veličina replike
 - n broj (ravnomjerno opterećenih) pomoćnih poslužitelja
- Opterećenje bez replikacije
 - $L1 = f_r \cdot (|_m + |_r)$
- Opterećenje s replikacijom temeljenom na dohvaćanju promjena sadržaja

• L2=
$$f_r \cdot I_m + n \cdot f_w \cdot I_r + (f_r - n \cdot f_w) \cdot I_m$$
 za $f_r > n \cdot f_w$

$$za f_r > n \cdot f_w$$

$$za f_r \le n \cdot f_w$$



Prosljeđivanje promjena sadržaja (1)

 Poslužitelji sadržaja prosljeđuju svim replikama promjene stanja sadržaja u trenutku nastanka promjene

Poslužiteljske Primjenjuju se od strane poslužitelja Poslužitelji trajnih trajnih replika prema korisnicima replike replika Korisnici



08.12.2020. Raspodijeljeni sustavi

Prosljeđivanje promjena sadržaja (2)

- Značajke primjene prosljeđivanja sadržaja
 - Ostvarivanje velikog stupnja konzistentnosti
 - Stvaranje dodatnog mrežnog prometa
 - Poslužitelji trajnih replika moraju imati zabilježene adrese svih replika koje obnavljaju i opis njihova stanja
 - U slučaju da neka od replika ukloni dio svojeg stanja, replika obavještava o promjenama stanja poslužitelja trajnih replika
 - Pogodno za korištenje u slučajevima s rijetkim izmjenama sadržaja replika i velikog broja korisnika



Prosljeđivanje promjena sadržaja (3)

- Oblici prosljeđivanja promjena sadržaja
 - Prosljeđivanje novog sadržaja
 - Može se proslijediti samo izmijenjeni dio sadržaja replike ili cjelokupni sadržaj replike
 - Koristi se kad je frekvencija čitanja puno veća od frekvencije pisanja
 - Prosljeđivanje obavijesti o promjenama sadržaja
 - Bolja je od prosljeđivanja novog sadržaja Koristi se kad je frekvencija promjene sadržaja (pisanja) puno veća od frekvencije čitanja, ali je lošija od dohvaćanja promjena sadržaja
 - Prosljeđivanje operacija za promjenu sadržaja
 - Postoje slučajevi kada se ne može primijeniti (npr. fotografije)
 - Zahtjeva dodatnu obradu na izvorištu i odredištu



Primjer [3]

- Parametri
 - l_o veličina poruke s operacijom za promjenu sadržaja
- Replikacija temeljena na prosljeđivanju novog sadržaja
 - L3 = $n \cdot f_w \cdot I_r$
- Replikacija temeljena na prosljeđivanju obavijesti o promjenama sadržaja

• L4 =
$$n \cdot f_w \cdot I_m + n \cdot f_w \cdot (I_m + I_r)$$
 za $f_r > n \cdot f_w$

• L4 =
$$n \cdot f_w \cdot I_m + f_r \cdot (I_m + I_r)$$
 za $f_r \le n \cdot f_w$

Replikacija temeljena na prosljeđivanju operacija za promjenu sadržaja

• L5=
$$n \cdot f_w \cdot l_o$$



Uspostava replikacije podataka

- Uvod u replikaciju podataka
- Organizacija sustava replika
- Razredba vrsta replika
 - Trajne replike, poslužiteljske replike, korisničke replike
- Održavanje konzistentnosti replika
 - Dohvaćanje promjena stanja replika
 - Prosljeđivanje promjena stanja replika
- Ostvarivanje operacija
 - Operacije pisanja sadržaja
 - Operacije čitanja sadržaja



Ostvarivanje operacija

 Održavanje konzistentnosti sadržaja replika tijekom provođenja operacija čitanja i pisanja od strane korisnika

- Obnavljanje stanja udaljenih replika
 - Zahtjev za obnavljanje stanja prosljeđuje se udaljenim replikama koje zajednički ostvaruju

- Lokalno obnavljanje stanja replika
 - Replike se dohvaćaju na računalo domaćin te se operacije provode lokalno



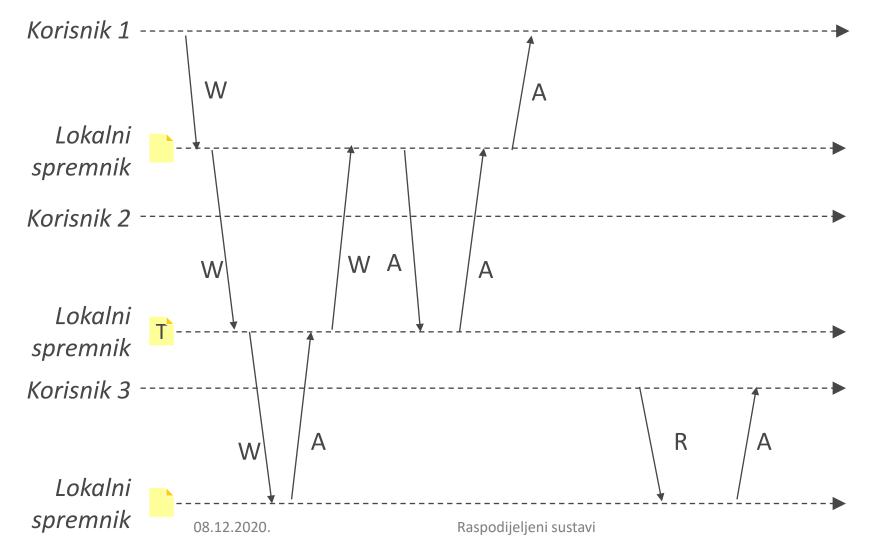
Obnavljanje stanja udaljenih replika (1)

- Model ostvarivanja operacija čitanja i pisanja
 - Zahtjevi za obnavljanje stanja prosljeđuju se udaljenoj trajnoj replici za zadani sadržaj
 - Udaljena trajna replika ostvaruje lokalnu promjenu sadržaja i prosljeđuje zahtjev za pisanje svim ostalim replikama u sustavu
 - Ostale replike nakon promjene sadržaja šalju potvrde te zatim trajna replika prosljeđuje potvrdu korisniku
 - Operacije čitanja se provode na lokalnoj replici ili bilo kojoj drugoj replici u sustavu replika



Obnavljanje stanja udaljenih replika (2)

W – Pisanje, R – Čitanje sadržaja, A – Rezultat, F – Dohvat replike





Obnavljanje stanja udaljenih replika (3)

- Značajke modela obnavljanja stanja udaljenih replika
 - Omogućava uspostavu slijedne konzistentnosti obzirom da samo glavna replika provodi operacije pisanja
 - Svi korisnici doživljavaju jednak redoslijed izvođenja operacija pisanja u vremenu bez obzira putem koje replike dohvaćaju sadržaj
 - U slučaju velikog broj pomoćnih replika, izvođenje operacije pisanja može zahtijevati značajnu količinu vremena potrebu za provođenje operacije pisanja na svim pomoćnim replikama



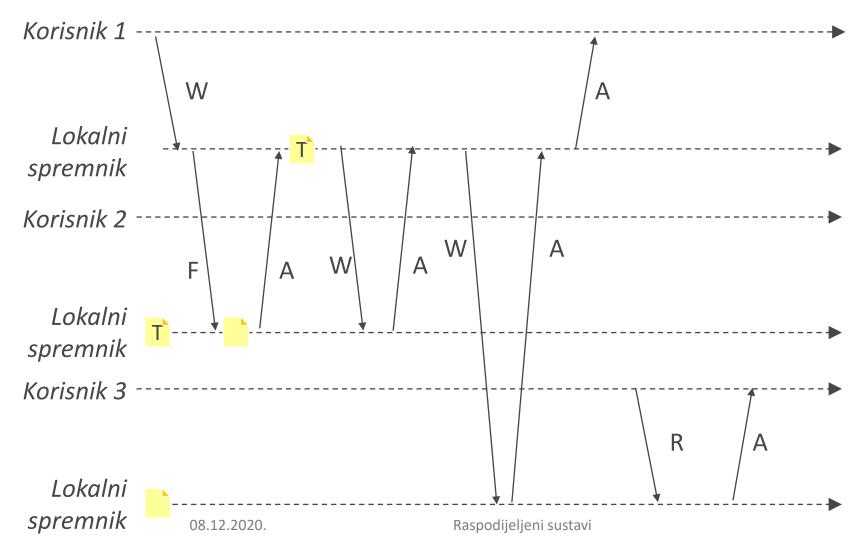
Lokalno obnavljanje stanja replika (1)

- Model ostvarivanja operacija čitanja i pisanja
 - Trajna replika se dohvaća na računalo domaćin
 - Provodi se operacija pisanja
 - Promjene ostvarene na trajnoj replici u lokalnom spremniku prosljeđuju se svim ostalim replikama u sustavu
 - Operacije čitanja se provode na lokalnoj replici ili bilo kojoj drugoj replici u sustavu replika



Lokalno obnavljanje stanja replika (2)

W – Pisanje, R – Čitanje sadržaja, A – Rezultat , F – Dohvat replike





Lokalno obnavljanje stanja replika (3)

- Značajke modela lokalnog obnavljanja stanja replika
 - Uzastopne operacije pisanja mogu biti provedene u kratkom vremenu na računalu domaćinu
 - Rezultati provođenja uzastopnih operacija mogu biti agregirani u jednu operaciju pisanja koja se provodi na pomoćnim replikama u raspodijeljenoj okolini
 - Korisnici koji čitaju sadržaj mogu pristupiti vlastitim lokalnim replikama neovisno o trajnoj replici



Usporedba modela replikacije u sustavima

	Baza podataka	P2P	WWW
Upravljanje	Centralizirano	Raspodijeljeno	Centralizirano
Povezanost	Čvrsta	Labava	Čvrsta
Raspoređivanje	Centralizirano	Raspodijeljeno	Centralizirano
Vrsta većine operacija	Pisanje i čitanje	Čitanje	Čitanje
Pouzdanost	Predvidiva	Nepredvidiva	Predvidiva
Raznorodnost sredstava	Ne	Da	Ne



62 od 66

Dodatne informacije (1)

Knjige

- S. Tanenbaum, M. van Steen: "Distributed Systems: Principles and Paradigms", Pretence Hall, 2002. (Poglavlje: Consistency and Replication)
- H. Attya, J. Welch: "Distributed Computing: Fundamentals, Simulations, and Advanced Topics", Wiley, 2004. (Poglavlje: Distributed Shared Memory)



Dodatne informacije (2)

- Znanstveni radovi
 - S. Goel, S. Buyya: "Data Replication Strategies in Wide-Area Distributed Systems", u: Enterprise Service Computing: From Concept to Deployment (ur. R. Qiu), pp.211-241, IGI Global, 2006.
 - http://www.buyya.com/papers/DataReplicationInDSChapter2006.pdf
 - P. Padmanabhan, L. Gruenwald, A. Vallur, M. Atiquzzaman: "A Survey of Data Replication Techniques for Mobile and Ad-Hoc Network Databases", Journal of Very Large Data Bases, Vol. 17, pp. 1143-1164, 2008.
 - http://www.cs.ou.edu/~database/documents/VLDB08.pdf



Pitanja za učenje i ponavljanje

- Objasnite vezu između replikacije i konzistentnosti.
- Objasnite način replikacije u HDFS-u.
- Objasnite poštuje li se slijedna konzistentnost u slijedu izvođenja operacija prikazanom na slajdu 17.
- Objasnite poštuje li se povezana konzistentnost u slijedu izvođenja operacija prikazanom na slajdu 20. ukoliko pretpostavimo da je operacija pisanja podatka B (uzročno) povezana s operacijom pisanja podatka A.
- Skicirajte slijedni dijagram korištenja poslužiteljskih replika.
- Skicirajte slijedni dijagram lokalnog obnavljanja stanja replike.



Pitanja za učenje i ponavljanje - nastavak

• 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_n=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.

