



Diplomski studij

**Informacijska i
komunikacijska tehnologija
Telekomunikacije i
informatika**

**Računarstvo
Računarska znanost
Programsko inženjerstvo i
informacijski sustavi**

Raspodijeljeni sustavi

Pitanja za provjeru znanja
2. blok predavanja

Ak.g. 2010./2011.

Napomena:

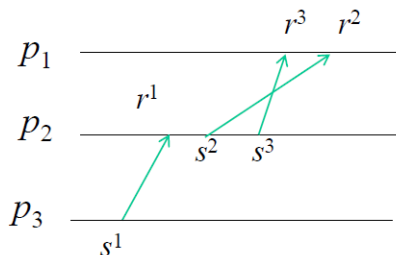
Preporučena literatura su bilješke s predavanja.

Zadatak 1 Objasnite za koje je od sljedeća tri svojstva raspodijeljenih sustava značajna komunikacijska složenost algoritama: a) replikacijska transparentnost b) skalabilnost c) otvorenost.

Komunikacijska složenost algoritama 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.

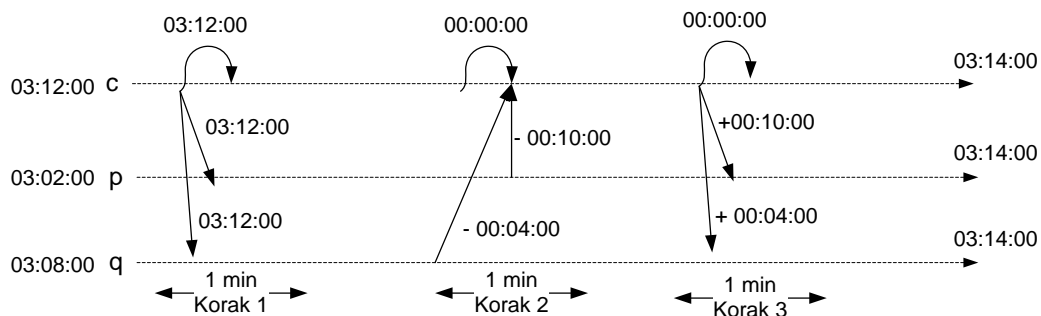
Zadatak 2 Objasnite model komunikacijskog kanala koji se temelji na uzročnoj slijednosti. Uzročna slijednost (causal ordering) osigurava da uzročno povezani događaji slanja dviju poruka istom primatelju rezultiraju primanjem u slijedu kojim su poslani.

Zadatak 3 Objasnite zašto za sljedeći primjer vrijedi CO ili vrijedi non-CO?



Za primjer vrijedi non-CO zato što proces p_1 prima poruke od p_2 drugačijim redoslijedom od redoslijeda slanja, a pri tome je slanje poruke 3 uvjetovano slanjem poruke 2.

Zadatak 4 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 minutu.



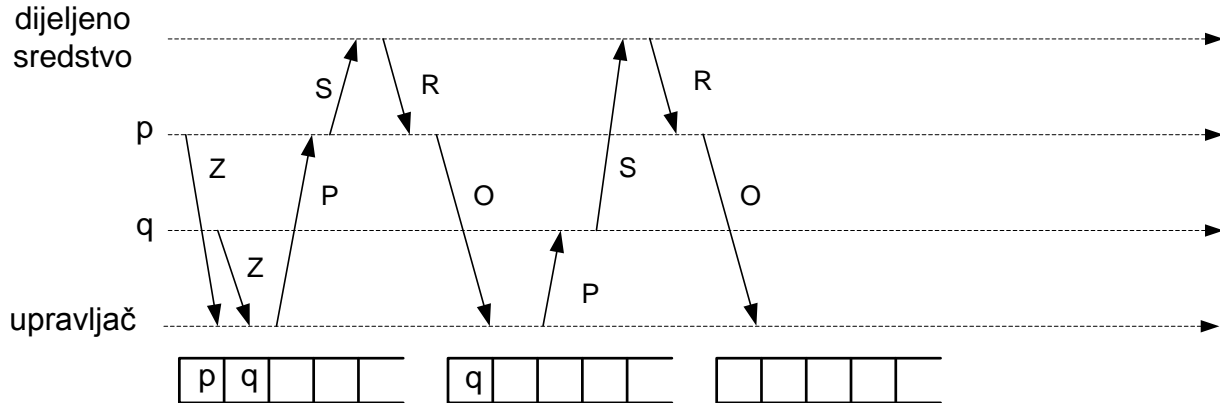
Korak 1:
Upravitelj šalje poruku s trenutnim vremenom svim računalima.

Korak 2:
Poslane poruke putuju 1 minutu i nakon primitka poruka, računala odgovaraju s porukom koja sadrži razliku lokalnog vremena u odnosu na primljeno vrijeme.

Korak 3:
Nakon primitka poruka odgovora, upravitelj šalje poruke zahtjeva koje sadrže vremenski pomak za svako računalo. Poruke zahtjeva putuju 1 minutu te nakon primitka poruke zahtjeva, svako računalo usklađuje lokalni satni mehanizam.

Zadatak 5 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

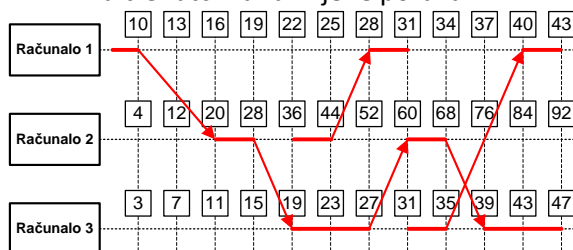


Proces p šalje zahtjev za zauzimanje sredstva,
zahtjev se sprema u rep
Proces q šalje zahtjev za zauzimanje sredstva,
zahtjev se stavlja u rep
Kako je zahtjev od R0 stigao prije, upravljač šalje
potvrdu R0 i uklanja njegov zahtjev iz repa

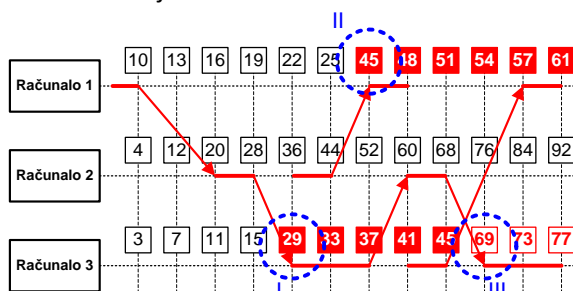
Proces p provodi operaciju pisanja
Proces p prima potvrdu
Proces p šalje poruku Upravljaču i otpušta pristup
Upravljač šalje poruku dojavu procesu q te mu
dodjeljuje pristup dijeljenom spremniku. Iz
repa zahtjeva uklanja se zahtjev od procesa p
Proces q provodi operaciju pisanja
Proces q prima potvrdu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

Zadatak 6 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.

1. Određivanje vrijednosti satnih mehanizama u trenutcima razmjene poruka



2. Primjena skalarnih oznaka vremena



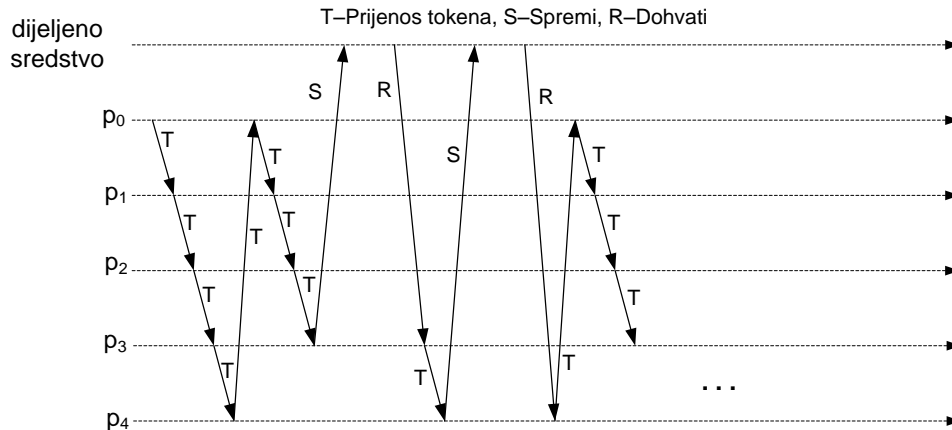
Trenutak I: Računalo 3 prima poruku od računala 2 s oznakom vremena $T_p=28$ koja je veća od lokalne oznake vremena $T_l=19$. Lokalni sat se pomiče na vrijednost $T_p+1=29$.

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 $T_p+1=69$.

Zadatak 7

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. Prikažite 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.

Zadatak 8

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.

Zadatak 9

Navedite i opišite značajke tri osnovna razreda replika podataka u raspodijeljenim sustavima.

U raspodijeljenim sustavima koriste se sljedeća tri razreda replika:

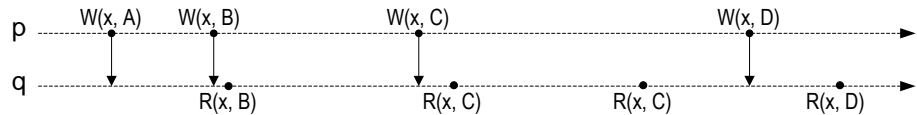
1. **Trajne replike:** Osnovni primjerci podataka koji su trajno postavljeni na poslužiteljima. Njihove postavke su statičke i upravljane od strane vlasnika podataka. Nad njima se najčešće ostvaruju operacije čitana podataka i poslužuju primjenom grozdova poslužitelja i replika poslužitelja.
2. **Poslužiteljske replike:** Poslužitelji podataka stvaraju nove replike trajnih replika i raspoređuju ih na dostupne poslužitelje u mreži. Odabir i raspoređivanje replika ostvaruje se u stvarnom vremenu tijekom rada poslužitelja.
3. **Korisničke replike:** Postupak repliciranja iniciran je odstrane korisnika putem korisničkih programa. Korisnički programi u lokalne spremnike spremaju pribavljene podatke. Potrebno je održavati konzistentnost lokalnog spremnika s poslužitelje od kojeg su podatci dohvaćeni.

Zadatak 10

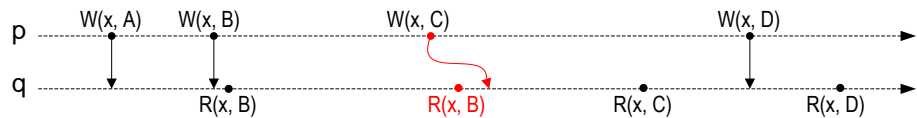
Objasnite što je stroga konzistentnost operacija u raspodijeljenim sustavima? Na primjeru procesa p i q prikažite slijed operacija čitanja i pisanja koji je a) u skladu i b) nije u skladu s načelima stroge konzistentnosti.

Model nalaže da su sve operacije pisanja trenutno vidljive svim ostalim procesima u raspodijeljenoj okolini. Naime, bez obzira koliko je kratko vremena prošlo od posljednje operacija pisanja, ako se provede operacija čitanja na bilo kojem računalu rezultat će biti upravo sadržaj zapisan u prethodnoj operaciji pisanja.

a)



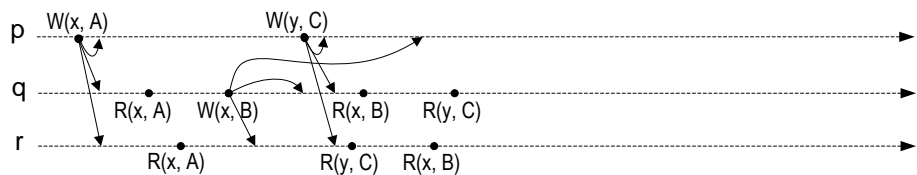
b)

**Zadatak 11**

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.

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.

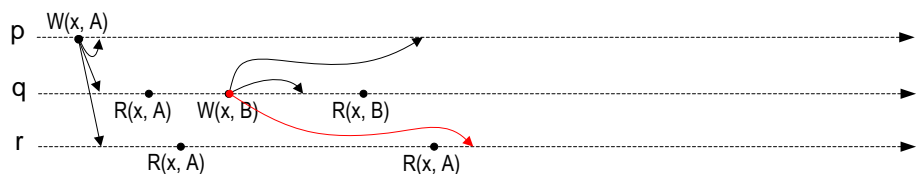
a)



Čitanje q: A->B->C

Čitanje r: A->C->B

b)

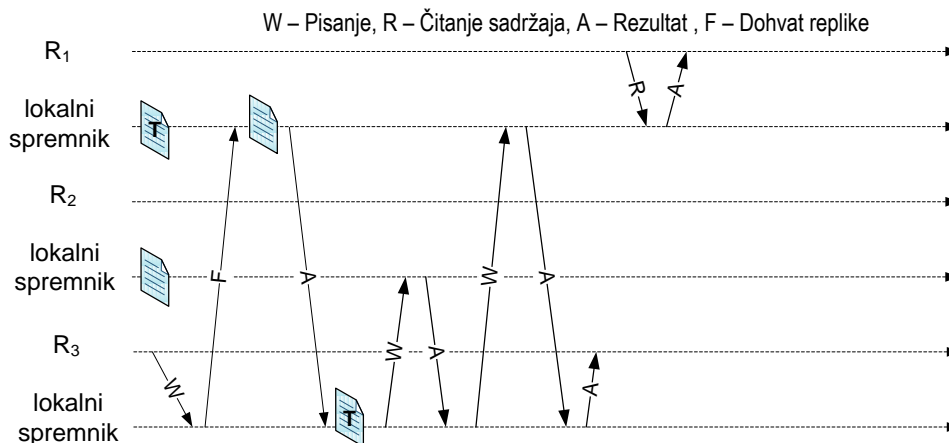


Čitanje q: A->B

Čitanje r: A->A

Zadatak 12 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.

R_3 upućuje zahtjev za provođenje operacije pisanja lokalnom spremniku sustava replika. Obzirom da to računalo u lokalnom spremniku ne sadrži trajnu repliku, računalo dohvaća trajnu repliku s R_1 u svoj lokalni spremnik. Replika na R_1 se pretvara u običnu repliku. Nakon prenošenja trajne replike u lokalni spremnik od R_3 , obnavlja se njeno stanje i proslijeđuje se ostalim računalima u sustavu. Nakon što ta računala pošalju potvrdu, R_3 proslijeđuje potvrdu korisniku koji je započeo operaciju pisanja. Ostali korisnici zatim imaju mogućnost provođenja operacije čitanja nad najbližom replikom u sustavu replika.



Zadatak 13 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 pretpostavite da korisnike isključivo poslužuju pomoćni 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 $l_p=1$ kB, $l_o=50$ kB i $l_r=100$ kB. Usporedite dobivena opterećenja s centraliziranim slučajem kada korisnike poslužuje glavni poslužitelj.

- PUSH metoda s proslijeđivanjem obavijesti o promjenama

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

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.

- PUSH metoda s proslijeđivanjem operacija za promjenu sadržaja

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

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.

- PUSH metoda s proslijeđivanjem cjelokupnog sadržaja replika

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

Nakon svake promjene, glavni poslužitelj šalje novu verziju replike svakom pomoćnom poslužitelju.

- PULL metoda

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

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.

- Centralizirani slučaj

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

Glavni poslužitelj odgovara replikom na svaki korisnički upit.

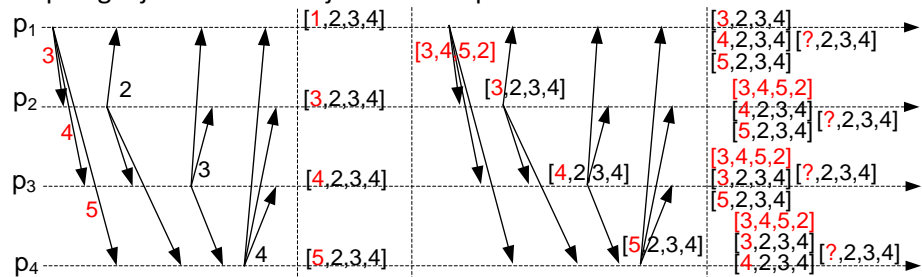
Zadatak 14 Objasnite razliku između ispada sustava i neispravnosti u sustavu.

Ispad sustava je stanje sustava koje se detektira kroz nemogućnost 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.

Zadatak 15 Pretpostavite 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?

$$k = 2, N = 3k + 1 = 7.$$

Zadatak 16 U grupi od 4 procesa (p_1, p_2, p_3 i p_4) proces p_1 je neispravan (pretpostavite 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.



Zadatak 17 Pretpostavite da se u grupi procesa s ispadima poštuje načelo virtualne sinkronosti. Proces p šalje poruku m grupi procesa G . Tijekom isporuke poruke m dogodi se ispad procesa p . Što se događa isporukom poruke m ?

Isporuka se prekida, procesi ignoriraju primljenu poruku.

- Zadatak 18** Pretpostavite da procesi P1 i P2 šalju poruke koje se isporučuju procesima P3 i P4 prema tablici. Navedite koju vrstu pouzdane komunikacije podržava grupa procesa P1, P2, P3 i P4?

Proces P1	Proces P2	Proces P3	Proces P4
šalje m11	šalje m21	prima m11	prima m11
šalje m12	šalje m22	prima m21	prima m12
šalje m13		prima m22	prima m21
		prima m12	prima m13
		prima m13	prima m22

Ova grupa procesa podržava pouzdani FIFO multicast.

- Zadatak 19** Navedite obilježja pouzdane komunikacije grupe procesa pod nazivom atomic multicast.

Atomic multicast je pouzdani virtualno sinkroni multicast s potpuno uređenim slijedom poruka.

- Zadatak 20** Objasnite razliku protokola *three-phase commit* u odnosu na *two-phase commit*.

Three-phase commit je sličan protokolu *two-phase commit* osim što koordinador 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, koordinador šalje GLOBAL_COMMIT.