

ISPIT IZ PREDMETA DISTRIBUIRANI SISTEMI

1. Objasniti razliku između tranzijentnih i perzistentnih komunikacija u distribuiranom sistemu baziranom na middleware sa razmenom poruka.

Odgovor je na prezentacijama Komunikacija u DS, slajdovi 62,63 i 64.

2. Tri računara A, B i C komuniciraju koristeći protokol koji implementira Lamportove vremenske markice. Na početku u svim računarima vrednosti vremenskih markica su 0. Kasnije nastupaju sledeći događaji:

- A šalje poruku M1 u B: „Zdravo“
- Nakon što je primio M1, B šalje poruku M2 u C: „A mi se javio“
- Nakon što je C primio M2, šalje poruku M3 u A: „B mi dosađuje“

- (a) Uz svaku od poslanih poruka napisati vrednost vremenske markice sa kojom je poslata:

Send (M1, 1)

Send (M2, 3)

Send (M3,5)

- (b) Nakon ovih događaja, poslate su još tri poruke:

- Nakon što je primio poruku M3, A šalje poruku M4 u B : „C je nedosledan“
- Nakon što je primio poruku M4, B šalje poruku M5 u A: „C mi dosađuje“
- A prima poruku M5.

Nakon što su sve poruke poslate i primljene, kolike su vrednosti vremenskih markica u svakom od računara A, B i C?

A	10
B	9
C	5

- (c) Da li opisani slučaj predstavlja primer relativne ili potpuno uređene komunikacije?

U pitanju je relativno uređenje događaja, tj. nema multicast-a.

3. a) Definirati pojmove defekt (fault), greška (error) i otkaz (failure) i navesti primer koji ilustruje ova tri pojma.

Odgovor je u prezentacijama od 27.04-30.04, slajd 3.

- b) Distribuirani sistem se sastoji od 4 replicirana servera. Pouzdanost svakog pojedinačnog servera je 0.9.

- i. Ako je sistem projektovan tako da može da funkcioniše ako je bilo koji od servera u funkciji, kolika je pouzdanost celog sistema?

Odgovor je u prezentacijama od 27.04-30.04, slajdovi 7 i 8. Pouzdanost sistema je $P=1-(0.1)^4=0.9999$

ii. Ako je sistem projektovan tako da može da funkcioniše samo ako su sva četiri servera u funkciji (ispravna), kolika je pouzdanost celog sistema?

Pouzdanost sistema je $P=(0.9)^4=0.6561$

4. U distribuiranom sistemu klijent koristi RPC za komunikaciju sa serverom. Klijent šalje zahtev serveru (npr. operation $x()$) sve dok ne dobije odgovor od servera.

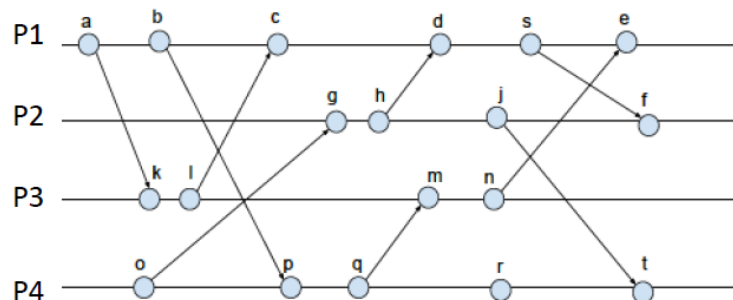
a. Koja RPC semantiku u slučaju greške je u ovom slučaju implementirana?

U pitanju je at least one semantika jer klijent šalje zahtev sve dok ne primi odgovor od servera.

b. Objasniti kako mora da bude implementiran server da bi se ovakava semantika obezbedila? Navesti bar dve mogućnosti.

Dve su mogućnosti na strani servera: Da implementira idempotentnu proceduru *operation $x()$* ili da server bude statefull i da vodi računa o klijentskim zahtevima i prepoznaje duplikate zahteva i u tom slučaju vrši samo retransmisiju poruke u kojoj je odgovor (ne izvršava ponovo proceduru *operation $x()$*)

5. Na slici su prikazana 4 distribuirana procesa koji međusobno komuniciraju. Prikazati konzistentni presek koji uključuje događaj **j** i **najmanji mogući** broj događaja u ostalim procesima.

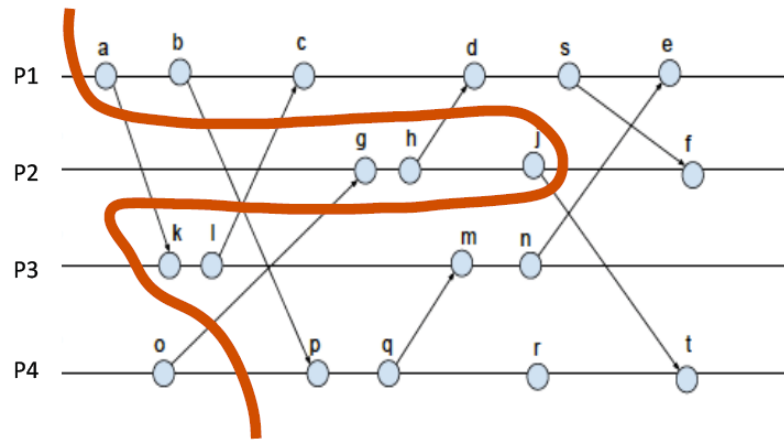


Rešenje:

Uslov koji mora biti zadovoljen da bi imali konzistentni presek je da ako je u nekom procesu zabeležen prijem neke poruke iz drugog procesa, tada mora konzistentnim presekom biti zabeležen trenutak slanja te poruke.

U konkretnom primeru, događaj **g** predstavlja prijem poruke iz procesa P4, pa konzistentni presek mora da obuhvati događaj **o** u procesu P4.

Events in
the cut:
g,h,j,o



Predmetni nastavnik