

# Odgovori na prvi kolokvijum - Računarstvo u Oblaku

## 1. Uvod

1. Šta je računarstvo u oblaku?
  - Računarstvo u oblaku je rezultat decenijskog napredka računarstva. Aplikacije su bazirane na klijent server arhitekturi gde je većina logike na serveru.
2. Koje sve resurse nudi na iznajmljivanje računarstvo u oblaku?
  - Serveri za računanje ili storage, usluge, mreža, aplikativni softver.
3. Koji su to rani modeli?
  - Grid computing i Utility computing.
4. Glavne karakteristike?
  - Grid computing: povezuju i koordinišu se geografski dislocirani računari koji imaju zajednički cilj.
  - Utility computing: podskup cloud computinga, korisnici računarske resurse skaliraju po svojim potrebama, korisnici unajmaju resurse poput procesora, skladišta ili čak i hardvera.
5. Koje su ključne prednosti računarstva u oblaku?
  - Resursi se dele između korisnika, resursi se multipleksiraju (bolja iskorišćenost jer neće dva korisnika u isto vreme imati ogromno opterećenje datog resursa), udruživanje resura, eliminacija početnih troškova, smanjenje troškova, elastičnost (brza adaptacija na ogroman skok potrebe za resursima), prilagođenost korisnicima.
6. Izazovi računarstva u oblaku?
  - Dostupnost (šta ako cloud provajder ne može više da pruža resurse), privatnost, težak prelazak na drugu platformu zbog velikih specifičnosti platforma, bottleneck u prenosu podataka, nepredvidive performanse (posledica deljenja resursa), teško upravljanje resursima na strani cloud provajdera.
7. Koja su tri glavna modela isporuke ponuđenih usluga (delivery models)?
  - SaaS, PaaS, IaaS.
8. Koje su osobine svakog od ovih modela isporuke?
  - SaaS je neki aplikativni softver sa biznis logikom koji se pruža krajnjim korisnicima. Korisnik nema kontrolu servera van aplikacije koju koristi.
  - Na PaaS-u je instaliran neki OS i korisniku se pruža nekakva vrsta platforme, a korisnik bira koje će instalirati aplikacije.
  - IaaS je je usluga u okviru koje se obezbeđuje infrastruktura i korisnik ima kontrolu nad operativnim sistemom.

## 2. Paralelno i distribuirano računarstvo

9. Koji koncepti razvijeni za paralelno i distribuirano računarstvo su ključni i za računarstvo u oblaku?
  - Mnogo aplikacija na cloudu moraju da obrađuju jako mnogo podataka, pa im treba moći paralelnog računarstva u vidu više instanci. Komunikacioni protokoli su jako vezani za distribuirano računarstvo. Snapshots. Upotreba monitora.
10. Osnovni koncepti paralelnog računarstva?

- Paralelni posao je generalno često efikasniji nego serijski posao, razbijanje na manje podprobleme.

11. Čime je ograničeno maksimalno ubrzavanje koje se paralelizacijom može postići? Amdalov zakon.

- Amdalov zakon kaže da ceo sistem može da benefitira ubrzanjem jednog dela, maksimalno onoliko koliko se taj deo koristi u sistemu tj. koliko ukupnog vremena se troši na taj deo.

12. Da li povećanje broja procesorskih jedinica garantuje stalno povećanje performansi sistema?

- Ne nužno, zato što se ne može sve paralelizovati a i zato što nekad paralelizacija košta više nego što ubrzava sistem (zbog sinhronizacije, headrooma paralelizma).

13. Koji su ključni izazovi/problemi paralelnih sistema? Šta tačno predstavlja svaki od njih?

- Velik headroom, race condition, deadlock, livelock, priority inversion (proces manjeg prioriteta izbací proces većeg prioriteta).

14. Koji su tipični modeli komunikacije između procesa?

- Razmena poruka (Golang) i deljenje memorije.

15. Na kojim nivoima možemo posmatrati paralelizaciju?

- Paralelizam na nivou bita, paralelizam na nivou instrukcija, paralelizam na nivou ciklusa (svaka iteracija for loopa se izvršava paralelno) i paralelizam na osnovu zadatka (problem = nezavisni zadaci koji se izvršavaju paralelno).

16. Koje su tipične arhitekture paralelnih sistema?

- SISD, SIMD, MIMD.

17. Šta su distribuirani sistemi?

- Distribuirani sistem je kolekcija računara povezanih preko mreže koji komuniciraju i softver za upravljanje tj. koordinaciju njih.

18. Glavne karakteristike distribuiranih sistema?

- Autonomnost komponenata, postoji više tačaka upravljanja, postoji više tačaka otkaza, resurs ne mora biti dostupan sve vreme, skaliranje je horizontalno, visok nivo dostupnosti i u lošim uslovima.

19. Koja poželjna svojstva treba da ima distribuirani sistem?

- Transparentan pristup tj. da se svim objektima pristupa putem istih operacija. Transparentnost lokacije tj. da korisnik ne zna gde su objekti. Transparentnost konkurentnog izvršavanja tj. da više procesa može da se izvršava konkurentno pristupajući deljenim objektima bez ometanja. Transparentnost replikacije tj. da se koriste nekakve vrste replika a da toga nisu svesni korisnici.

20. Globalno stanje grupe procesa? Koji model sa koje dve ključne apstrakcije se koriste za sagledavanje distribuiranih sistema?

- Za lakše razumevanje distribuiranih sistema se koristi model apstrakcije koja se zasniva na dve ključne komponente: procesi i kanali komunikacije.

21. Šta su procesi i niti?

- Proses je neki program, a nit je manji deo tog programa.

22. Šta su komunikacioni kanali?

- Komunikacioni kanal je sredstvo putem kog procesi komuniciraju slanjem poruka. Postoji događaj slanja i događaj primanja.

23. Koja je ključna prepostavka koju moramo uzeti u obzir kada koristimo komunikacione kanale?

- Komunikacioni kanali su nepouzdani.

24. Kako sagledavamo stanje procesa?

- Proces je opisan svojim stanjem u datom momentu. Stanje sadrži sve informacije potrebne da se proces restartuje.

25. Kako posmatramo stanje samog komunikacionog kanala?

- Stanje komunikacionog kanala između dva procesa, koje mora biti usmereno, se sastoji od poruka koje je prvi proces poslao, a drugi nije primio.

26. Šta predstavlja protokol?

- Protokol je konačni skup poruka između procesa, koje razmenjuju da bi se koordinisali.

27. Kako sagledavamo globalno stanje grupe procesa?

- Globalno stanje grupe procesa je unija svih stanja procesa i svih stanja komunikacionih kanala.

28. Koji događaji definišu globalno stanje grupe procesa?

- Komunikacioni događaji.

29. Kako broj mogućih tranzicija utiče na složenost analize/debagovanja sistema?

- Što je broj mogućih tranzicija veći tj. što je rešetka većih dimenzija, to je debagovanje teže.

30. Zašto su neophodni mehanizmi za otkrivanje i uklanjanje grešaka u komunikacionim kanalima?

- Otkrivanje i ispravljanja grešaka otkazanog komunikacionog kanala zanemaruje to što su kanali nepouzdani i onda procesi mogu da komuniciraju pouzdano i u nepouzdanim kanalima.

31. Svaki proces ima lokalnu istoriju promena povezanih sa lokalnim vremenom? Šta je problem sinhronizacije?

- Da. Kada procesi komuniciraju, njihove poruke mogu biti izgubljene i onda je kasnije nemoguće sinhronizovati satove tih procesa pa time ni redosled akcija.

32. Da li je moguća apsolutna globalna suglasnost o vremenu u velikim distribuiranim sistemima?

- Nije moguća, jer lokalni satovi konstantno driftuju od prave vrednosti vremena.

33. Da li je apsolutni konsenzus o tačnom vremenu neophodan ili je za koordinaciju sistema potreban mehanizam koji omogućava utvrđivanje redosleda događaja?

- Nije potrebno znati tačno vreme za svaku akciju tj. poruku, ali je potrebno znati redosled akcija tj. poruka i taj redosled nam omogućava logički sat LC.

34. Zašto je bitan koncept uzročno-posledičnih veza (causality)?

- Distribuirani sistemi se generalno analiziraju i dizajniraju na principu znanja koja akcija je izazvala koju drugu akciju, pa je jako bitan koncept uzročno-posledičnih veza.

35. Koji događaji se smatraju konkurentnim?

- U okviru distribuiranih sistema, događaji koji u globalnoj istoriji nemaju vezu su konkurentni događaji.

36. Šta je koncept logičkog sata i kako rešava problem nepostojanja konsenzusa o globalnom vremenu?

- Logički sat LC je alat u distribuiranim sistemima koji definiše redosled događaja i on je apstrakcija za globalno vreme (koje ne može da postoji). Mapira događaje na celobrojnu vrednost na osnovu formule:

$$LC(e) = \begin{cases} LC+1 & \text{- ako je } e \text{ lokalni događaj} \\ \max(LC, TS(m))+1 & \text{- ako je } e \text{ događaj prijema poruke} \end{cases}$$

Ovo znači da će vrednost "vremena" za sledeći lokalni događaj biti samo "vreme + 1", a vrednost za sledeći prijem poruke će biti veći broj između "lokalnog vremena" i "vremena udaljenog računara" povećan za 1.

37. Koji problemi mogu nastati pri prenosu poruka nepouzdanim kanalima?

- Kada prenosimo poruke nepouzdanim kanalima, mogu stići u pogrešnom redosledu.

38. Šta je FIFO isporuka poruka?

- FIFO (First In First Out) metod isporuke garantuje da će poruke biti isporučene onim redosledom kojim su poslate.

39. Šta je Causal delivery isporuka poruka?

- Causal delivery je proširenje FIFO metoda isporuke tako da važi da poruke budu isporučene redosledom kojim su poslate i u situacijama kada proces može primiti poruke iz različitih izvora.

40. Šta su stabilne poruke?

- Ako želimo da imamo stabilne poruke, one moraju imati timestamp-ove. Stabilna poruka je ona kojoj vrednost timestamp-a nije manja od poruka od poruka koje trebaju da stignu posle nje.

41. Koje pravilo važi kod Consistent message delivery?

- Procesi imaju pristup globalnom vremenu, nema driftovanja lokalnih satova; kašnjenje poruka nije veće od  $\delta$  i svaka poruka ima timestamp.

42. Šta je monitor (ili monitoring) proces?

- Monitor proces je proces koji je zadužan da formira sliku globalnog stanja tako što kontaktira sve pojedinačne procese.

43. Šta su run, cut, frontier of the cut i consistent cut?

- Run: torka svih događaja u globalnoj istoriji koja je konzistenta sa svakom lokalnom istorijom procesa. Cut - podskup lokalnih istorija svakog procesa. Frontier of the cut - torka koja sadrži sve poslednje događaje u svim procesima koji su u tom rezu. Consistent cut - cut koji je obuhvatio sve događaje koji zadovoljavaju uzročno posledičnu vezu.

44. Šta je konkurentnost i koje izazove donosi?

- konkurentnost je kada se više stvari izvršavaju u isto vreme (prividna paralelnost ako je računar dovoljno brz), ali ne nužno paralelno.

45. Šta su atomične aktivnosti i kako (u kojim oblicima) ih je moguće obezbediti?

- Atomične aktivnosti su zadaci koji se moraju izvršiti u celini (nedeljive operacije), tranzakcione. Da bi se atomičnost koristila, mora biti obezbeđena od strane hardvera.

46. Šta su i zašto su bitni protokoli za ostvarivanje konsenzusa?

- Protokoli za ostvarivanje konsenzusa su bitni jer je i sam konsenzus bitan a to je proces postizanja saglasnosti o validnosti date alternative kada ih ima više.

47. Od kojih prepostavki se polazi u osnovnom Paxos protokolu?

- Procesi se izvršavaju na procesorima i komuniciraju preko mreže i svi oni mogu doživeti otkaze. Procesi rade na proizvoljnim brzinama i imaju stabilan storage pa se mogu pridružiti ponovo kada otkažu. Mreža može izgubiti, duplirati ili premestiti redosled podacima a poruke se šalju asinhrono a vreme isporuke može biti proizvoljno dugo.

48. Koji su ključni učesnici (uloge) u Paxos protokolu?

- Client, Proposer, Acceptor, Learner, Leader.

### 3. Virtuelizacija

49. Šta je virtuelizacija?

- Virtuelizacija je kreiranje virtuelne verzije nečega. Za softversko inženjerstvo, ovaj termin se najčešće primenjuje za OS-ove, mrežne resurse i skladišni prostor.

50. Koji je značaj virtuelizacije u računarstvu u oblaku?

- Izolacija performansi, bezbednost sistema, bolje performanse i pouzdanost.

51. Na koje načine se može vršiti virtuelizacija?

- Multipleksiranje je kada se kreira više virtuelnih objekata na osnovu jednog fizičkog (primer: više procesa koristi jedan procesor). Agregacija je suprotno od multipleksiranja gde se od više fizičkih objekata kreira jedan virtualni (primer: RAID kod diskova). Emulacija je kada se kreira virtualni objekat jednih karakteristika koristeći fizički objekat drugih karakteristika (primer: virtuelizacija OS-a, SSD emulira RAM).

52. Šta je uloga VMM/hipervizora?

- Hypervisor je softver koji omogućava da se virtuelizovanom objektu da pristup delu hardverskih resursa. Može se instalirati kao aplikacija na već postojećem OS-u ili direktno na hardver.

53. Šta su različiti nivoi interfejsa (ISA, ABI, API)?

- API je način da naš program zove biblioteke koje služe za olakšani razvoj datog problema. ABI je način da program, ili češće biblioteka, zove procedure operativnog sistema. ISA (Instruction Set Architecture) su instrukcije procesora koje se direktno izvršavaju na hardveru i najčešće ih OS sam zove.

54. Kakva je razlika između hipervizora tipa 1 i tipa 2?

- Tip 1 se instalira direktno na hardver, a tip dva se instalira kao aplikacija na već postojećem OS-u.

55. Koje ključne operacije VMM mora obavljati da bi uspešno virtuelizovao procesore i memoriju?

- Hvatanje privilegovanih instrukcija, hvata interapte i prosleđuje ih gostujućim OS-evima, održava paging tabelu, nadgleda performanse sistema.

56. Koji su uslovi za uspešnu virtuelizaciju?

- Program / OS koji VMM kontroliše se mora ponašati isto kao i da je pokrenut normalno. VMM ima potpunu kontrolu nad virtuelizovanim resursima. Skoro sve instrukcije se moraju izvršavati bez VMM intervencije.

57. Virtuelizacija CPU x86 arhitekture I izazovi koje donosi?

- x86 arhitektura ima prstene privilegije. Nulti prsten je obično OS, a treći prsten su obično korisničke aplikacije. Potrebno je pomeriti VMM na 0. prišten, a OS na 1. prsten.

58. Šta je puna virtuelizacija i koje su njene prednosti/mane?

- Puna virtuelizacija je tehnika virtuelizacije gde se gostujući OS izvršava neizmenjen. Prednosti: radi bez hardverske podrške virtuelizacije, OS ostaje neizmenjen, izolacija i bezbednost. Mane: sporo.

59. Šta je paravirtuelizacija i koje su njene prednosti/mane?

- Kod paravirtuelizacije, kritični pozivi kernela OS-a koji virtualizujemo su zamjenjeni sa hiper pozivima (direktnim pozivima VMM-a). Prednost: brže, manji overhead. Mane: smanjena portabilnost.

60. Šta je hardverski potpomognuta virtuelizacija i njene prednosti/mane?

- VMM se izvršava ispod nivoa 0, a OS na nivou 0. Kritični pozivi tj. instrukcije se automatski hvataju od strane hipervizora čime se uklanja potreba prevođenja kod paravirtuelizacije. Prednosti: jako brzo. Mane: nije napisano ali nešto poput potrebna hardverska podrška u procesoru.

61. Šta je sve moguće virtuelizovati?

- Mrežu, storage, serverske resurse, podatke, desktope, aplikacije...

## 4. Infrastruktura cloud providera

62. Koje ključne entitete identificuje NIST model cloud computinga?
- Korisnik servisa, ponuđač servisa, prenos, posrednik, revizor.
63. Od čega se sastoji globalna infrastruktura AWS-a?
- Od snažnih računarskih platforma.
64. Koji su ključni servisi AWS-a?
- EC2 serveri, SQS messaging, CloudWatch za nadgledanje, Management Console za upravljanje zakupljenim servisima, S3 za storage fajlova, EBS i razne baze.
65. Koje grupe servisa nudi Google?
- Dosta ML IaaS i SaaS servisa i u skorije vreme dosta PaaS servisa poput podrške za upravljanje kontejnerima. Google App Engine, Container Engine, Cloud Storage, Cloud Virtual Network.
66. Koje su ključne komponente MS Azure platforme?
- Compute, Storage, Fabric Controller, CDN, Aplikacije i podaci.

## 5. AWS fundamentals

67. Šta je AWS S3 i čemu služi?
- AWS S3 je object based (objekat = fajl) usluga za čuvanje fajlova. Služi za čuvanje binarnih fajlova bez neke strukture. Ključ je ime fajla, a vrednost bajtovi tog fajla.
68. Šta je AWS IAM i čemu služi?
- AWS IAM omogućava upravljanje korisnicima u okviru jedne organizacije na AWS-u. Pod upravljanje spada dodeljivanje i oduzimanje prava na Management Konzoli, politika izmene lozinke, multifactor autentikačija, ...
69. Šta je AWS Cloud Front?
- AWS Cloud Front je Amazonov CDN. Omogućava da se sadržaj korisnicima isporučuje preko distribuirane mreže u zavisnosti od korisnikove geografske lokacije. Osnovna jediica je Edge Location što je server na kome se kešira sadržaj.
70. Koje zone identificuje ANSI/TIA-942-2005 standard u data centrima?
- **Ne postoji odgovor u prezentacijama.**
71. Koji slojevi se tipično uspostavljaju u hijerarhijskoj organizaciji mreže data centara?
- **Ne postoji odgovor u prezentacijama.**
72. Koja je razlika između ToR i EoR pristupa u organizaciji access sloja?
- **Ne postoji odgovor u prezentacijama.**

## 6. Upravljanje resursima

73. Šta je upravljanje resursima? Na koje kriterijume za evaluaciju sistema direktno utiče?
- Upravljanje resursima je osnovna funkcija sistema. Uticaje na cenu, performanse i funkcionalnost.
74. Da bi se postiglo efikasno upravljanje resursima u cloudu kakva optimizacija je neophodna?
- Višeciljna optimizacija gde se upravljanje ne sme fokusirati samo na jednu stvar.
75. Jedan od primarnih ciljeva upravljanja resursima u oblaku je obezbeđivanje elastičnosti. Koji je ključni izazov sa kojim se upravljanje resursima suočava u ovakvim okruženjima?

- Veliko, fluktuirajuće opterećenje.

76. Koja su dva osnovna pristupa (načina) upravljanju resursima?

- Predviđanje opterećenja i alociranje u napred, i korišćenje autoscaling algoritma koji može raditi samo ako postoje slobodni resursi i samo ako je monitoring sistem dobar pa će znati koji su to resursi koji mogu da se realociraju.

77. Zašto je centralizovani pristup upravljanju resursima tema polemika od nastanka cloud okruženja?

- Jer se smatra da centralizovana kontrola teško može da se izbori sa brzim i nepredvidivim promenama i da obezbedi kontinualnu isporuku.

78. Koje su osnovne grupe politika upravljanja resursima u oblaku? Šta je cilj svake od njih?

- Kontrola prihvatanja (sprečava sistem da prihvati rad na zadacima zbog neke definisane politike); Alokacija kapaciteta (kako se alociraju resursi); Balansiranje opterećenja (kako učiniti da je opterećenje jednako podeljeno na sve instance servera); Optimizacija utroška energije (kako minimizovati potrošenu energiju); Quality of Service (kako ispuniti sve zahteve iz ugovora).

79. Šta je problem raspoređivanja (scheduling) u računarskom sistemu?

- Odlučivanje kako dodeliti resurse sistema različitim zadacima i korisnicima.

80. Šta je "rasporedivač" (scheduler) i koja je njegova funkcija?

- Rasporedivač je program koji implementira algoritam raspoređivanja resursa.

81. Koje su ključne odluke koje rasporedivač donosi u pogledu resursa?

- Količini resursa koju je potrebno dodeliti i trajanju te dodele resursa nekom zadatku ili korisniku.

82. Šta su ključne željene karakteristike algoritma za raspoređivanje resursa?

- Efikasnost, pravednost, sprečavanje gladi.

83. Koji tipovi vremenskog ograničenja za izvršavanje zadataka se najčešće identificuju? Za koje aplikacije je primerena određena vrsta vremenskog ograničenja?

- Postoje tvrda i meka ograničenja, gde tvrda ograničenja označavaju zadatke koji moraju da se završe u nekom roku i postoje penali ako se ne završe, a meka ograničenja su zadaci koji ne moraju nužno u tom roku da se završe i nemaju penale. Aplikacije multimedije imaju meka vremenska ograničenja, a aplikacije u realnom vremenu imaju tvrda vremenska ograničenja.

84. Kada se radi raspoređivanje na VM, koje dodatne faktore treba uzeti u obzir?

- Postoji veći broj raznovrsnih zadataka, konačan broj zadataka se može obavljati na svakom serveru, zadaci mogu imati drugačije potrebe u smislu vremena za izvršavanje, performanse, izolacija resursa između različitih zadataka.

85. Šta je Proportional Share raspodela resursa?

- Neki zadaci zahtevaju više CPU vremena pa im je dodeljeno proporcionalno više.

86. Šta je Fair Share raspodela resursa?

- Svaki zadatak bi trebalo da dobije istu količinu CPU vremena.

87. Šta je Work Conserving, a šta Non Work Conserving režim rada rasporedivača?

- Kod Work Conserving moda, cilj je održati procesor u radu pa će CPU nastaviti biti korišćen dok god ima posla. Kod Non Work Conserving moda, cilj je da je svakom zadatku dodeljen udio i kad svaki zadatak potroši udio, CPU prestaje sa radom.

88. Šta je Preemptive, a šta Non Preemptive metod funkcionisanja rasporedivača?

- Kod Preemptive moda CPU-a, rad jednog zadatka može biti prekinut i preraspodela resursa se radi kada dođe novi zadatak. Kod Non Preemptive moda CPU-a, rad jednog zadatka ne može biti prekinut i preraspodela resursa se radi kada zadatak završi sa korištenjem CPU-a.

89. Šta je Workload management (WM)? Koja su dva osnovna pristupa WM?

- Workload management je sistem koji aplikacijama precizno dodeljuje resurse bilo koje vrste. Aplikacije govore koje performanse su im potrebne, a WM dodeljuje resurse na osnovu toga. Postoji staticki WM (resursi se aplikaciji dodeljuju jednom) i dinamički WM (resursi se prilagođavaju trenutnim potrebama aplikacije).

90. Šta je BVT algoritam?

- Borrowed Virtual Time. Pravična raspodela, dodeljuje CPU vreme zadacima sa najmanjom vrednošću virtuelnog vremena. Preemptive, WC. Ne podržava NWC pa ograničava izolaciju.

91. Šta je SEDF algoritam?

- Simple Earliest Deadline First. Svaki zadatak specificira vremenski odsečak, period i da li može dobiti ekstra vreme (WC mod). Preemptive, WC i NWC. Pravičnost zavisi od dužine perioda. Nedostaje globalno balansiranje na multiprocesorskim okruženjima.

92. Šta je Credit Scheduler algoritam?

- Obezbeđuje da ni jedan fizički CPU nije u idle režimu ako ima stvarnog posla na virtuelnim CPU-ovima. Za svaku VM, rasporediće dodeljuje težinu i limit. Limit je  $0 = WC$  režim rada za tu VM. Limit nije  $0 = NWC$  režim rada za tu VM. Svaki virtuelni CPU dobija 30ms za izvršavanje pre nego što biva zamenjen. Non preemptive, WC i NWC. Globalno balansiranje na multiprocesorskim okruženjima.