

Osnove distribuiranih sistema i distribuiranog programiranja

Seminarski rad u okviru kursa
Metodologija stručnog i naučnog rada
Matematički fakultet

Stefan Isailović, Marko Stanković, Darko Vasiljević, Lazar Ristić
mi16029@alas.matf.bg.ac.rs, mr12296@alas.matf.bg.ac.rs,
mi16449@alas.matf.bg.ac.rs, mi16150@alas.matf.bg.ac.rs

8. maj 2021.

Sažetak

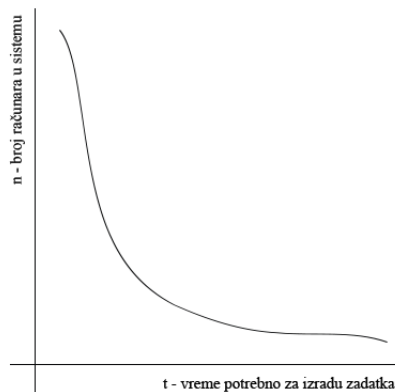
Sa porastom broja podataka, raste i potreba za brzinom obrade tih podataka. Recimo da imamo hiljadu podataka za obradu; umesto da sve te podatke šaljemo na jedan računar i čekamo obradu, bolje je iste te podatke podeliti na, recimo, 10 jednakih računara. Na taj način bismo, u idealnom slučaju, imali iste podatke obrađene 10 puta brže. Za ovakav način rada su zaduženi distribuirani sistemi. Kao i svaki sistem, tako i distribuirani ima svoje karakteristike, vrste, probleme i rešenja za probleme.

Sadržaj

1	Uvod	2
2	Distribuirani sistemi i njihove karakteristike	2
2.1	Karakteristike distribuiranih sistema	2
2.2	Organizacija distribuiranih sistema	3
3	Topologije distribuiranih sistema	4
3.1	Klijent-server arhitektura	4
3.2	Troslojna arhitektura	5
3.3	Višeslojna arhitektura	5
3.4	Peer-to-peer arhitektura	6
4	Problemi i rešenja distribuiranog programiranja	7
4.1	Kompleksna arhitektura i otpornost	7
4.2	Oporavak čvorova i dostupnost	7
4.3	Usporenje	8
4.4	Isporučivanje	8
5	Platforme distribuiranog programiranja	8
6	Zaključak	9
	Literatura	10

1 Uvod

Prvi distribuirani sistem javio se 70ih godina prošlog veka u vidu lokalne mreže, kao potreba za bržom i lakšom komunikacijom. Od tada, pa do danas, distribuirani sistemi su se poboljšavali da bi omogućili brži i lakši rad, kao i bržu i lakšu komunikaciju. Na slici 1 možemo videti zavisnost vremena t za izradu zadatka od broja računara n u sistemu. Što je više računara u sistemu između kojih se posao deli, to je vreme za izradu zadatka kraće.



Slika 1: Grafik zavisnosti t od n

2 Distribuirani sistemi i njihove karakteristike

Definicija 2.1 *Distribuirani sistem je skup nezavisnih računara koji korisnicima izgleda kao koherentan sistem [2].*

Veličina distribuiranog sistema se može kretati od nekoliko uređaja, do miliona računara. Računari funkcionišu konkurentno, a ukoliko dođe do kvara, ne odražava se na ostale računare sistema. Zajednička mreža na koju su povezani može biti bežična ili žična, ili kombinacija obe.

Definicija distribuiranih referiše na dva osnovna principa distribuiranih sistema:

1. Distribuirani sistem je kolekcija uređaja koji mogu funkcionisati nezavisno.
2. Koherentnost sistema čini da klijent ima osećaj da koristi jedan računar, a ne kompletan sistem računara.

2.1 Karakteristike distribuiranih sistema

Svaki distribuirani sistem opisan je određenim karakteristikama, dobrim i lošim. Ključne prednosti distribuiranih sistema su skalabilnost, otpornost na grešku, dobar odziv i efikasnost. Ključne mane distribuiranih sistema su: obrada grešaka, bezbednost i cena.

- *Skalabilnost* (eng. scalability): Distribuirani sistem omogućava horizontalno skaliranje¹, čime se omogućava više saobraćaja u sistemu. Skaliranje nije ograničeno brojem uređaja, pa se iz tog razloga smatra najbitnijom prednošću.
- *Otpornost na grešku* (eng. partition tolerance/fault tolerance): Ukoliko dođe do kvara jednog uređaja, ne utiče na ceo sistem.
- *Dobar odziv* (eng. availability): Zbog velikog broja računara koji mogu da odgovore na zahtev korisnika, kašnjenje odgovora je retko.
- *Efikasnost* (eng. efficiency/consistency): Velika količina podataka se razbija u sitnije delove i daje na obradu računarima u sistemu. Na ovaj način podaci se brže obrađuju, jer više računara rešava zajednički problem.
- *Obrada grešaka* (eng. error handling): Iako je prednost što svi računari u distribuiranom sistemu funkcionišu nezavisno jedan od drugog, problem može nastati prilikom pada jednog računara koji bi trebalo da je jednak sa ostalima u sistemu; zbog čega je na tom računaru došlo do greške, a na ostalima ne?
- *Bezbednost* (eng. security): Zbog velikog broja korisnika (računara) i automatski velikog broja podataka, bezbednost celog sistema mora biti na visokom nivou. Ti podaci prolaze kroz više računarskih mreža, koje su ponekad i javne, pa su samim tim izloženiji napadima (krađi). Najpoznatija vrsta napada je *DoS* (eng. Denial of Service) [3].
- *Cena* (eng. cost): *S obzirom na to da distribuirani sistem može biti ogroman, samim tim i cena tog sistema može biti ogromna.*

Iako bi bilo idealno da distribuirani sistem ima sve ove karakteristike, u realnom svetu je teško dizajnirati takav sistem, o čemu više govori CAP teorema.

CAP teorema

Termin CAP teorema prvi put se pojavio 2000. godine, kada ju je formulisao Erik Bruer (eng. Eric Brewer)[6], profesor na Univerzitetu Berkli i bivši potpredsednik Guglove infrastrukture.

CAP teorema je ključna teorema u teoriji distribuiranih sistema koja govori da distribuirani sistem ne može istovremeno biti efikasan, otporan na greške i imati dobar odziv [5].

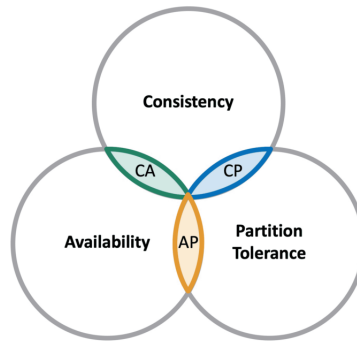
U najvećem broju slučajeva, korisnik je ostavljen da bira koje dve stvari će njegov sistem istovremeno imati. Dobar odziv je svojstvo koje se uvek gleda kao prioritet, jer korisnik želi garanciju da će dobiti odgovor na zahtev, makar to bilo i sa zakašnjenjem.

2.2 Organizacija distribuiranih sistema

Kao što je već navedeno, distribuirani sistem čini skup uređaja i oni mogu funkcionisati nezavisno jedan od drugog. Uređaj u sistemu nazivamo **čvor**.

Čvorovi funkcionišu tako da postignu neki zajednički cilj, a međusobno komuniciraju slanjem poruka. Zbog organizacije kolekcije čvorova, distribuirani sistem je često implementiran kao *prekrivena mreža* (eng. overlay network) [7]. U ovakvom okruženju, čvor sadrži listu ostalih čvorova sa kojima je komunikacija moguća.

¹Horizontalno skaliranje se postiže dodavanjem novih uređaja u sistem, dok se vertikalno skaliranje postiže dodavanjem više resursa (RAM memorije, procesora, itd.)



Slika 2: Grafički prikaz CAP Teoreme

Prekrivena mreža može biti strukturirana i nestruktuirana. **Strukturirana prekrivena mreža** je mreža u kojoj svaki čvor ima definisanu listu svojih suseda - čvorova sa kojima može da komunicira. **Nestruktuirana prekrivena mreža** je mreža u kojoj svaki čvor ima određen broj suseda s kojima nasumično uspostavlja konekciju.

Najpoznatiji tip prekrivene mreže je **P2P** (eng. *peer-to-peer*) o kom će biti više reči u narednoj glavi.

Drugi princip distribuiranih sistema govori o koherentnosti sistema. Sistem se može nazvati koherentnim ukoliko se ponaša u skladu sa očekivanjima korisnika. Konkretno dobra koherentnost znači da korisnik ne može da ustanovi na kom računaru iz sistema se operacija trenutno izvršava. Takođe, korisnika ne bi trebalo da interesuje ni gde se podaci skladište, kao ni način njihove obrade. Ovo se naziva *transparentnost distribuiranja* (eng. *distribution transparency*) [7].

3 Topologije distribuiranih sistema

Distribuirani sistemi mogu da poseduju različite vrste topologija i arhitektura, ali uglavnom se mogu svrstati u jednu od sledeće četiri kategorije[4]:

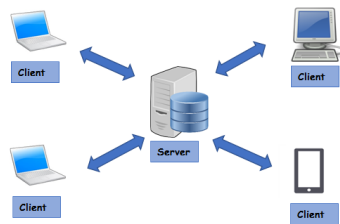
- Klijent-server arhitektura
- Troslojna arhitektura
- Višeslojna arhitektura
- Peer-to-peer arhitektura

U nastavku ćemo obraditi svaku od ovih arhitektura i upoznati se detaljnije sa njihovim osobinama.

3.1 Klijent-server arhitektura

Arhitektura klijent-server je struktura koja vrši podelu zadataka između servera - strana koja deli svoje resurse i/ili usluge, i klijenta - strana koja traži usluge. Server i klijent vrše komunikaciju preko mreže. Na slici 3 prikazan je diagram klijent-server arhitekture.

Ideja se zasniva na tome da server (host) omogući deljenje resursa pomoću serverskih aplikacija, a da klijent po potrebi koristi deljenje resurse.



Slika 3: Dijagram klijent-server arhitekture

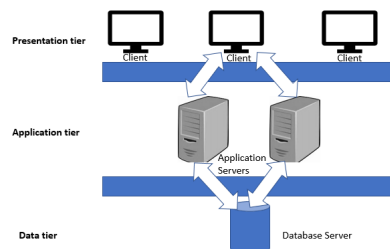
Tipični primer aplikacija koje koriste klijent-server arhitekturu su e-mail, web strane, network printing itd.

3.2 Troslojna arhitektura

Troslojna arhitektura je specijalan tip klijent-server arhitekture, samo za razliku od nje, ova arhitektura je podeljena na tri sloja:

- Prezantacioni sloj - predstavlja korisnički interfejs kojim se postiže komunikacija korisnika sa ostalim slojevima.
- Aplikacioni sloj - sadrži biznis logiku celog sistema.
- Sloj podataka - sloj koji se sastoji od baze podataka ili neke druge strukture za skladištenje i pristupanje podacima.

Na slici 4 prikazan je diagram troslojne arhitekture



Slika 4: Dijagram troslojne arhitekture

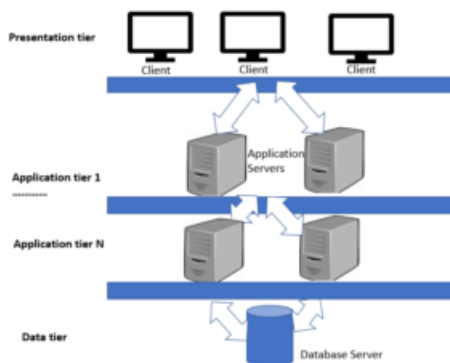
Ova struktura je nastala sa namerom da se omogući fleksibilnost prilikom razvoja aplikacija, jer je ovaj model pogodan za nezavisno razvijanje svakog sloja bez remećenja funkcionalnosti ostalih slojeva.

Primer 3.1 *Tipičan primer aplikacija ovog tipa su aplikacije koje postoje na različitim platformama u vidu aplikacija za desktop, mobilne uređaje (telefon, tablet) i kao aplikacija koja može da se koristi kroz web pretraživač (na primer, Netflix aplikacija za gledanje video sadržaja).*

3.3 Višeslojna arhitektura

Ovaj tip arhitekture predstavlja najopštiji tip arhitekture distribuiranih sistema. Ideja je da se aplikacija podeli na proizvoljan broj slojeva

koji su zaduženi za procesiranje, obradu podataka i prezentaciju, koji mogu biti razdvojeni fizički i logički. Na slici 5 prikazan je dijagram višeslojne arhitekture. Benefiti korišćenja strukture ovakvog tipa:

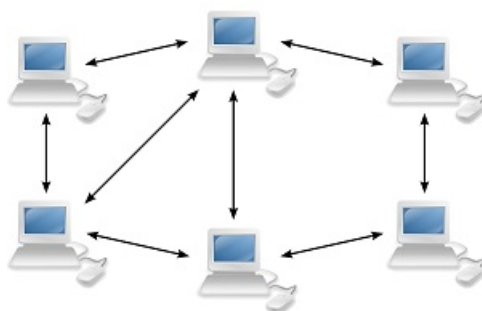


Slika 5: Dijagram višeslojne arhitekture

- Sigurnost - mogućnost da se svaki od slojeva zaštiti odvojeno ne nužno istim metodama.
- Lakoća održavanja - mogućnost dodavanja i menjanja odvojenih slojeva bez uticaja na postojeće.
- Skalabilnost - mogućnost dodavanja novih resursa po slojevima bez uticaja na ostale slojeve.
- Fleksibilnost - mogućnost proširenja svakog pojedinačnog sloja na bilo koji način.

3.4 Peer-to-peer arhitektura

Peer-to-peer arhitektura predstavlja drugačiji tip arhitekture distribuiranih sistema od do sada pomenutih arhitekture. Naime, ova arhitektura se zasniva na čvorovima koji ostvaruju interakciju jedni sa drugima i tako podjednako učestvuju u deljenju podataka i resursa. Na slici 6 prikazan je dijagram Peer-to-peer arhitekture.



Slika 6: Dijagram peer-to-peer arhitekture

Neke od bitnih karakteristika ove arhitekture su:

- Svi čvorovi (računari) ove mreže čuvaju podatke individualno i omogućavaju odgovarajuću vrstu sigurnosti na istom nivou.
- Svi čvorovi ove mreže istovremeno i pružaju i koriste resurse. Za razliku od klijent-server arhitekture, koja se sa povećanjem broja čvorova opteretiti, ovaj tip mreže povećava svoj kapacitet dodavanjem novih čvorova.
- Uzimajući u obzir da je ovaj tip mreže istovremeno i server i klijent, teško je obezbediti adekvatnu sigurnost za sve čvorove.
- Mnogi operativni sistemi kao što su Windows i Mac OS poseduju podršku za implementaciju ovog tipa mreže.

4 Problemi i rešenja distribuiranog programiranja

Pored prednosti koje donose u odnosu na sisteme zasnovane na monolitnim aplikacijama, distribuirani sistemi sa sobom nose i razne probleme. Osvrnucemo se na neke od važnijih i njihovo uobičajeno rešavanje u praksi, uz poređenje sa monolitnim sistemima.

4.1 Kompleksna arhitektura i otpornost

Jedan od ciljeva distribuiranog rešenja je velika otpornost na kvarove. Iz perspektive postavljanja arhitekture, treba razmišljati da svaka komponenta sistema može da se pokvari. Da bi sistem bio otporan, svaka od komponenti bi trebalo da ima mogućnost replikacije. što znači da arhitektura sistema treba biti takva da se za svaki deo poslovne logike koju obavlja jedan mikro servis, može podići više instanci. Što je veći broj instanci/replika to je sistem otporniji.

Na primer, serverska mašina na kojoj se nalazi instanca nekog mikro servisa može da ostane bez struje. Iz ugla arhitekture softverskog rešenja nije pouzdano oslanjati se na podršku koju u takvom slučaju pruža hosting provajder. Potrebna je takva postavka da jedan mikro servis može da se replicira i u slučaju pada jedne instance, ostale instance mogu da preuzmu opterećenje. Ako bismo otišli korak dalje, broj aktivnih instanci jednog mikro servisa može da zavisi od trenutnog opterećenja.

Ovakvi zahtevi, samim tim što se ne tiču direktno poslovne logike, usložnjavaju arhitekturu. Složenija arhitektura povlači veći posao u održavanju sistema, ali to se rešava tako što se posao deli po timovima, tako da je jedan tim specijalizovan za jedan deo sistema.

4.2 Oporavak čvorova i dostupnost

Iako možemo imati veliki broj instanci iste aplikacije kao podršku, ipak moramo brinuti i o tome šta će se dešavati u slučaju pada. Takođe, nije jednostavno utvrditi da li je instanca pala, ili se radi o mrežnom problemu pa se zbog toga dugo čeka na odgovor. U slučaju da se komunikacija sa jednim čvorom aplikacije odvija putem HTTP zahteva, možemo definisati najduže vreme čekanja na odgovor, nakon koga obustavljamo zahtev. Ako se koristi soket, jedno od rešenja bi moglo biti periodično slanje ping poruka ka instanci, da bi se utvrdilo stanje soket konekcije. Ako instanca odgovara na ovu poruku, to znači da je soket konekcija prisutna, a ako nema odgovora došlo je do prekida i konekcija se mora ponovo uspostaviti.

Obezbeđivanje visoke dostupnosti sistema je jedan od primarnih ciljeva programiranja distribuiranih sistema. Dostupnost sistema se definiše kao odnos vremena u kom sistem može da opsluži zahteve korisnika i vremenskog intervala po kom merimo.

Dostupnost	Trajanje zastoja u toku dana
90%	2,4 sata
99%	14,40 minuta
99.9%	1,44 minuta
99.99%	8,64 sekundi
99.999%	864 milisekundi

Tabela 2: Dostupnost sistema [8]

4.3 Usporenje

Pošto se poslovi dele između čvorova, između njih mora postojati komunikacija kako bi svaki čvor prosledio rezultat svog rada. Ova komunikacija se odvija preko mreže i to predstavlja jedan od nedostataka distribuiranih sistema. Ispravnost izvršavanja nekog zadatka zavisi od ispravnosti i propusnosti mreže, a važan faktor je i brzina slanja podataka. Kod distribuiranih sistema, koji lako skaliraju horizontalno, postoji određen nivo usporenja u odnosu na monolitne sisteme upravo zbog mrežne komunikacije. Međutim, kada se poveća pritisak na sistem, to usporenje neće biti gore, jer se samim dodavanjem novih čvorova povećava i propusnost sistema.

Postoji više načina kako redukovati neizbežno usporenje sistema. Jedan način je korišćenje keš memorije, međutim i to dolazi uz cenu. Ako imamo samo jedan keš server, onda taj server predstavlja usko grlo i kada on padne, nemamo zamenu. To često može biti dovoljno, u zavisnosti od zahteva aplikacije, jer će sistem raditi i bez keša kada recimo keš server padne, pa dok god njegov oporavak ne traje predugo sistem će raditi sporije, ali će ipak raditi. Više keš servera, sa druge strane, pridodaje kompleksnosti i može vrlo lako dovesti do nekonzistentnosti podataka.

4.4 Isporučivanje

Kada se radi isporuka nove verzije distribuiranog softvera, ako imamo n instanci servisa koje treba zameniti novom verzijom, mora postojati i neka automatizacija tog procesa jer nije praktično raditi zamenu „ručno“. Ovo u isto vreme predstavlja i izazov i pogodnost. Izazov u smislu definisanja i implementiranja procesa zamene, vraćanje unazad ukoliko dođe do neke greške itd. Pogodnost je u tome što se zamena može izvršiti postepeno tako da sistem sve vreme radi. Recimo, instance servisa se zamenjuju novim jedna po jedna, sve dok se ne zameni poslednja.

5 Platforme distribuiranog programiranja

Prve ideje o distribuiranom programiranju potiču još iz 1970.-tih godina, sa nastankom lokalnih mreža kao što je Ethernet. Ovaj tip sistema se kasnije preneo i na ARPANET, jednog od predaka današnjeg Interneta, kao i na e-mail servis, koji je predstavljao najraniji primer veće aplikacije

distribuiranog tipa.[1] Danas postoji veći spektar platformi i tehnologija koje koriste ili omogućavaju korišćenje distribuiranog načina rada:

- Internet - kao jedna od najvećih platformi.
- Telekomunikaciona mreža - započeta kao peer-to-peer mreža, doživela je veliki napredak razvijanjem VOIP (Voice over IP)
- Veštačka inteligencija - mogućnost da se iskoristi snaga računanja i paralelnog procesiranja sa ciljem da se obradjuju veliki skupovi podataka.
- Distribuirani real-time sistemi - mnoge industrije i kompanije, kao što je avionska, Uber i Lyft, koriste ovakav tip sistema, koji može da funkcioniše lokalno i globalno.
- Paralelno procesiranje - sa rastom operativnih sistema, procesora i cloud servisa, distribuirano programiranje je počelo da obuhvata i paralelno procesiranje.

6 Zaključak

Kroz ovaj rad pokazali smo značaj distribuiranih sistema, njihove benefite, kao i njihove mane. Sagledali smo sve vrste sistema, uočili ključne razlike i sličnosti među vrstama. Predstavili smo platforme ovih sistema i videli njihovu svrhu. Pokazali smo probleme i rešenja za iste. Dalji razvoj distribuiranih sistema može biti samo na bolje, s obzirom na to da će se razvojem moćnijih mašina u budućnosti, moći prevazići neki od trenutnih problema distribuiranih sistema. Budući da je podataka sve više, tako će i potreba za distribuiranim sistemima biti sve veća.

Literatura

- [1] M. Banks. *On the Way to the Web: The Secret History of the Internet and its Founders*. Apress, 2012.
- [2] Amanda Fawcett. What are distributed systems? a quick introduction, 2020. on-line at: <https://www.educative.io/blog/distributed-systems-considerations-tradeoffs#what>.
- [3] Mohamed Firdhous. Implementation of Security in Distributed Systems – A Comparative Study . *International Journal of Computer Information Systems*, 2(2), 2011.
- [4] Kothiwale N. Kamble A., Shinde T. Real Time And Distributed Computing Systems. *Journal of Computer Engineering*, 2017.
- [5] Royans.net. Brewer’s cap theorem on distributed systems, 2012. on-line at: <https://www.royans.net/2010/02/brewers-cap-theorem-on-distributed.html>.
- [6] Nancy A. Lynch Seth Gilbert. Perspectives on the CAP Theorem. *Perspectives on the CAP Theorem*.
- [7] M. van Steen and A. S. Tanenbaum. *Distributed Systems 3rd Edition*. Maarten van Steen, 2017.
- [8] Roberto Vitillo. *Understanding Distributed Systems*. Roberto Vitillo, 2021.