Analiza mrežne infrastrukture u funkciji koncepta Cloud Computing

Barišić, Gabriela

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:723862

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**



Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository







Sveučilište u Zagrebu

Fakultet prometnih znanosti

ZAVRŠNI RAD

ANALIZA MREŽNE INFRASTRUKTURE U FUNKCIJI KONCEPTA CLOUD COMPUTING

NETWORK INFRASTRUCTURE ANALYSIS IN CLOUD COMPUTING

Mentor: doc. dr. sc. Ivan Grgurević

Studentica: Gabriela Barišić
JMBAG: 0135237258

Zagreb, rujan 2017.

ANALIZA MREŽNE INFRASTRUKTURE U FUNKCIJI KONCEPTA CLOUD COMPUTING

SAŽETAK

Cloud Computing je nova paradigma čije su usluge promijenile način poslovanja današnjih tvrtki. Naime, on omogućava korisnicima pristup računalnim resursima bilo gdje i bilo kada. Kako bi se pružila određena razina usluge, *Cloud Computing* se sastoji od kompleksne mrežne infrastrukture. Tema ovog završnog rada je analiza potrebne pasivne i aktivne mrežne infrastrukture i tehnologija za ostvarenje pouzdane usluge *Cloud Computinga*. S obzirom da različiti javni davatelji, različito koriste svoju infrastrukturu, napravljena je usporedba njihovih usluga. Tako, korisnik može prema vlastitim potrebama odabrati najbolje rješenje.

KLJUČNE RIJEČI: *Cloud Computing*, mrežna infrastruktura, karakteristike, pohrana, poslužitelji

SUMMARY:

Cloud Computing is a new paradigm whose services change the way business is held within enterprises. It provides access to computing resources anywhere and anytime for users. In order to achieve level of service, Cloud Computing has complex underlying network infrastructure. The subject of this bachelor's thesis is analysis of passive and active network infrastructure and technology for reaching reliable services of Cloud Computing. Given that different public providers use their infrastructure in different ways, comparison of their services was made. Thus, user can choose the best solution based on their needs.

KEYWORDS: *Cloud Computing*, network infrastructure, characteristics, storage, servers

SADRŽAJ

1. Uvod ... 1
2. Razvoj koncepta Cloud Computing ... 3
2.1. Povijest ... 3
2.2. Defincija Cloud Computinga ... 3
2.3. Ugovor o razini usluge u okruženju Cloud Computing ... 5
3. Karakteristike koncepta Cloud Computing ... 7
3.1. Samousluga na zahtjev ... 7
3.2. Široki pristup mreži ... 7
3.3. Udruživanje resursa ... 7
3.4. Brza elastičnost ... 8
3.5. Mjerena usluga ... 8
3.6. Tehničke karakteristike ... 8
4. Mogućnosti primjene koncepta Cloud Computing ... 9
4.1. Modeli dostave Cloud Computinga ... 9
4.1.1. Softver kao usluga (SaaS) ... 10
4.1.2. Platforma kao usluga (PaaS) ... 10
4.1.3. Infrastruktura kao usluga (IaaS) ... 11
4.1.4. Podatci kao usluga (DaaS) ... 12
4.2. Razvojni modeli Cloud Computinga ... 12
4.3. Cloud federacija ... 13
4.4. Višekriterijska analiza Cloud Computing umrežavanja ... 14
5. Mrežna infrastruktura u konceptu Cloud Computing ... 16
5.1. Opći zahtjevi cloud infrastrukture ... 16
5.1.1. Skalabilnost ... 16
5.1.2. Konvergencija ... 17
5.1.3. Resursno usklađivanje ... 17
5.1.4. Performans ... 17
5.1.5. Dinamična migracija virtualne mašine ... 18
5.2. Radni okvir cloud infrastrukture ... 18
5.3. Mrežni model cloud infrastrukture ... 19
5.3.1. Funkcionalni zahtjevi za intra-cloud mrežu ... 20
5.3.2. Funkcionalni zahtjevi inter-cloud mreže ... 21
5.4. Pohrana ... 21

5.4.1. RAID 21
5.4.2. SAN 22
5.4.3. NAS 23
5.4.4. Hibridni sustav NAS i SAN 23
5.4.5. DAS 23
5.5. Poslužitelji 24
5.5.1. Mainframe poslužitelj 24
5.5.2. UNIX poslužitelji 24
5.6. Mrežni sloj Cloud Computinga 24
5.6.1. Mrežni izazovi u IaaS sloju 26
5.6.2. 10 Gb Ethernet 28
5.7. Virtualizacija infrastrukture Cloud Computinga 29
5.7.1. Hipervizor 29
5.7.2. Vrste virtualizacije 30
5.7.3. Virtualni privatni server 32
5.7.4. Virtualno umrežavanje 33
6. Analiza mrežne infrastrukture koncepta Cloud Computing 34
6.1. Amazon EC2 34
6.1.1. Analiza performansi Amazon EC2 36
6.1.2. Analiza Amazon EC2 u svrsi računalstva visokih performansi 37
6.2. Microsoft Azure 37
6.2.1. Azure usluge pohrane 38
6.2.2. Azure računalne usluge 39
6.3. Google App Engine 40
7. Zaključak 44
Literatura 45
Popis kratica i akronima 48
Popis ilustracija 51

1. Uvod

U novije vrijeme, optimizacija resursa i smanjenje troškova nad resursima je postala primarna za sve tvrtke. S toga, izgradnja potpune mrežne infrastrukture s obzirom na njenu neiskorištenost, je neisplativa. Na primjer, kapacitet i mogućnosti poslužitelja se iskorištavaju jako malo te većinom rade s malim opterećenjem. S toga je posljednjih godina sve popularnija virtualizacija i računalstvo u oblaku. Virtualizacija je rad više logičkih procesa na jednom fizičkom računalu uz podjelu njegovih resursa.

Upravo uz pomoć virtualizacije, omogućeno je postojanje *Cloud Computinga*. Ideja *Cloud Computinga* je jednostavan pristup svim računalnim uslugama krajnjeg korisnika putem Interneta. Korisnik se tereti prema količini korištenja usluge, a ima joj pristup bilo kada i bilo gdje. Korisnik može sam prema svojim potrebama rezervirati resurse *Cloud Computinga* i odabrati željenu uslugu. Dakle, postoje različiti modeli i vrste *Cloud Computinga*. Ipak, svima je zajednička neka vrsta mrežne infrastrukture koja omogućuje ponudu usluge, veliku razinu dostupnosti i pouzdanosti te skalabilnost usluga. Cilj ovog završnog rada je analiza i usporedba mrežne infrastrukture u funkciji koncepta *Cloud Computing*. Svrha završnog rada je utvrditi način iskorištenja mrežne infrastrukture za potrebe pružanja usluga temeljenih na *Cloud Computingu*. Rad je sastavljen od sedam cjelina, a to su:

1. Uvod
2. Razvoj koncepta *Cloud Computing*
3. Karakteristike koncepta *Cloud Computing*
4. Mogućnosti primjene koncepta *Cloud Computing*
5. Mrežna infrastruktura u konceptu *Cloud Computing*
6. Analiza mrežne infrastrukture koncepta *Cloud Computing*
7. Zaključak

U uvodu je predstavljena tema koja će biti obrađena u završnom radu. Definirani su cilj i svrha rada te su navedene teze s kratkim opisima.

Drugo poglavlje, nakon kratkog pregleda povijesnog razvoja *Cloud Computinga*, donosi definicije *Cloud Computinga* te osnovne tehnologije koje su utjecale na njegov razvoj te koje su i danas osnovni dio njega. Uz to, objašnjen je ugovor o razini usluge u konceptu *Cloud Computinga* kao bitna stavka za korisnike pri prelasku s tradicionalnog računalstva na usluge *Cloud Computinga*.

Treće poglavlje pobliže objašnjava osnovne karakteristike koncepta *Cloud Computing* koje je definirao *National Institute of Standard and Technology* – NIST, institut za standarde i tehnologiju u SAD-u, a to su: samousluga na zahtjev, široki pristup mreži, udruživanje resursa, brza elastičnost i mjerena usluga. Pored toga, objašnjene su i dvije tehničke karakteristike, labavo spajanje i tolerantnost na kvarove.

Četvrto poglavlje razrađuje mogućnosti primjene koncepta *Cloud Computing*. Naime, postoje tri osnovna modela dostave *Cloud Computinga* (softver kao usluga - SaaS, platforma kao usluga - PaaS i infrastruktura kao usluga - IaaS), a sve češće se spominje i novonastali

podatci kao usluga - DaaS. Nadalje, objašnjeni su razvojni modeli *Cloud Computinga* koji mogu biti: privatni *cloud, cloud* zajednica, javni *cloud* i hibridni *cloud.* Uz njih, objašnjena je *cloud* federacija i neki modeli višekriterijske analize odabira *Cloud Computing* usluge.

U poglavlju „Mrežna infrastruktura u konceptu *Cloud Computing*" definirani su opći zahtjevi *cloud* infrastrukture (skalabilnost, konvergencija, resursno usklađivanje, performanse i dinamične migracije procesa) te radni okvir *cloud* infrastrukture. Prikazan je mrežni model *cloud* infrastrukture te funkcionalni zahtjevi za intra-cloud mrežu i inter-cloud mrežu. Nadalje, objašnjeni su osnovni tipovi pohrane i servera, opisani su mrežni izazovi i zahtjevi te pred sam kraj objašnjena je virtualizacija i njene vrste.

U šestom poglavlju, dobiven je uvid u tri najveća javna davatelja usluge *Cloud Computinga*: Amazon EC2, Microsoft Azure i Google App Engine. Opisane su njihove glavne oznake te usluge. Na kraju, napravljene su i manje usporedbe njihovih performansi, prednosti i mane te sposobnost za rad u svrsi računalstva visokih performansi.

U zadnjem poglavlju završnog rada, *Zaključku*, sustavno su opisani najvažniji elementi istraživanja koji su opširnije elaborirani u radu. Na kraju rada nalazi se popis kratica i akronima, te popis slika i tablica koje su prikazane u završnom radu.

2. Razvoj koncepta *Cloud Computing*

U ovom poglavlju su obrađeni najrelevantniji aspekti razvoja *Cloud Computinga,* njegove definicije te vrste razine usluge koje zadovoljava. Riječ „*cloud*“ (oblak) označava apstrakciju osnovne infrastrukture (računala, mreže, podatkovne pohrane) koja omogućava normalan rad bilo kojeg *cloud computing* sustava [1].

2.1. Povijest

Može se reći da je razvoj *Cloud Computinga* započeo 50.-ih godina prošlog stoljeća s glavnim računalom. Više korisnika su bili u mogućnosti pristupiti centralnom računalu kroz jednostavne terminalne uređaje čija je jedina funkcija bila omogućavanje pristupa glavnom računalu. Zbog velikih troškova i održavanja glavnog računala, tada takav pristup nije bio praktičan za organizacije ili one jednostavno nisu trebale velike kapacitete pohrane i procesne moći koje je glavno računalo nudilo. Omogućavanje zajedničkog pristupa jednom resursu je rješenje koje je imalo ekonomskog smisla za sofisticiranu tehnologiju.

Oko 70.-ih godina prošlog stoljeća kreiran je koncept virtualnih mašina (VM). Korištenjem softvera za virtualizaciju omogućeno je pokretanje više operacijskih sustava simultano u izoliranom okruženju. Potpuno računalo može biti pokrenuto unutar jednog fizičkog hardvera koje može imati potpuno drugačiji operacijski sustav. Virtualne mašine su unaprijedile zajednički pristup glavnom računalu dozvoljavajući više različitih računalnih okruženja na jednom fizičkom okruženju.

90.-ih godina prošlog stoljeća telekomunikacijske tvrtke su počele nuditi virtualne privatne mreže – VPN (eng. *virtual private network*). Telekomunikacijske tvrtke su prije nudile jednu dodijeljenu s kraja na kraj podatkovnu vezu. Novost, virtualna privatna mreža, imala je jednaku razinu usluge kao i njihove dodijeljene usluge. Umjesto izgradnje fizičke infrastrukture kako bi više korisnika imalo vlastitu vezu, telekomunikacijske tvrtke su sada omogućile zajednički pristup istoj fizičkoj arhitekturi [2]. Nadalje, to je omogućilo poslužiteljima da nude jednaku količinu propusnosti pri nižoj cijeni tako da preusmjeravaju mrežni promet u stvarnom vremenu kako bi prilagodio iskorištenje mreže. Tako nije bilo moguće točno predvidjeti kojim će putem putovati podatci između poslužitelja i korisnika. Kao rezultat, poslužiteljeve mrežne odgovornosti su bile predstavljene u obliku oblaka koji simbolizira crnu kutiju s gledišta krajnjeg korisnika. U ovom smislu, termin „oblak“ u frazi „računalstvo u oblaku“ metaforički predstavlja Internet i njegovu osnovnu infrastrukturu [3].

NASA-ina *OpenNebula* je 2008. postao prvi *open-source* softver za implementaciju privatnih i hibridnih *cloud computing* sustava. Nedugo nakon toga pokrenute su mnogobrojne *Cloud Computing* usluge poput *Amazon Elastic Compute Cloud, Microsoft Azure* i *Google App Engine*. Dakle, 2000.-ih godina *cloud computing* je postao populariziran te se može donijeti konačna definicija [3].

2.2. Defincija *Cloud Computinga*

National Institute of Standard and Technology – NIST, institut za standarde i tehnologiju u SAD-u je definirao računalstvo u oblaku kao model koji omogućava sveprisutni, prigodan,

na zahtjev mrežni pristup dijeljenih konfiguriranih mrežnih resursa (npr. mreže, serveri, pohrana, aplikacije) koje mogu biti brzo opskrbljene i puštene s minimalnim upravljačkim trudom ili interakcijom uslužnog poslužitelja [4].

Ova definicija označava *cloud computing* kao model korisnosti ili „*pay as you go*“. Jednom kad se korisnik spoji, on koristi usluge koliko želi i kad želi, a naplaćuje mu se za korištene resurse. S obzirom da resursi preko kojih se dostavlja usluga, mogu biti podijeljeni između širokog broja korisnika, iskorištenje resursa i učinkovitost rada mogu biti više nego što bi bile za dodijeljene resurse za svakog individualnog korisnika. S time se također smanjuje cijena usluge za korisnika naspram alternativnih privatnih resursa za isporuku iste usluge [5].

Cloud computing je u mnogo načina konglomerat više različitih računalnih tehnologija i koncepata poput *grid computinga*, virtualizacije, autonomnog računalstva, servisno-orijentirane arhitekture – SOA (eng. *Service-oriented Architecture*), *peer-to-peer (P2P) computing* i sveprisutnog računalstva [3].

Grid computing je oblik distribuiranog računalstva koje koordinira i dijeli računalne, aplikacijske, podatkovne i mrežne resurse preko dinamične i geografski raspršene organizacije. Tako, *grid computing* tehnologije obećavaju promjene pri organizaciji kompleksnih računalnih problema [6]. *Grid computing* je primjer virtualnog računalnog resursa čiji je glavni cilj distribucija procesiranja jako kompleksnih ili vremenski iscrpljujućih aplikacija na grupu različitih uređaja kako bi se zadržali namijenjeni rezultati aplikacije u što kraćem vremenu. *GridGain* je otvorena *cloud* platforma za razvoj i pokretanje Java aplikacija. Ona može podijeliti inicijalno kompleksan zadatak na više zadataka. Ovi zadatci su dostavljeni različitim uređajima i svaki od zadataka se izvršava paralelno. Na kraju, procesni rezultati svih zadataka su spojeni u jedan krajnji rezultat. Problem povezan s *grid* sustavima je prijenosna barijera koja nastaje zbog različitih operacijskih sustava, kompajlera i sl. u računalnim uređajima koji tvore *grid* okruženje [1].

Virtualizacija je proces apstrakcije i enkapsulacije računalnih sredstava tako da se oni mogu koristiti na način koji odgovara određenoj primjeni. Virtualizirati se mogu računalne mreže, programi i operacijski sustavi. Virtualizacijom se postiže bolja iskorištenost računalnih infrastruktura jer se omogućuje njihovo istovremeno korištenje u različitim sustavima. Ova tehnologija odvaja aplikacije i operacijske sustave od nižeg sloja softvera. Kao rezultat, gostujući operacijski sustav kao i aplikacije pokrenute na gostujućem operacijskom sustavu su ponuđene s virtualnom mašinom. Virtualizacija tehnologije poboljšava razinu iskoristivosti modernih hardvera servera i omogućava integraciju višestrukih aplikacija na jako mali prostor fizičke mašine [7].

Servisno-orijentirana arhitektura (SOA) je dizajnirana paradigma koja pokušava kombinirati poslovne procese pakirane kao automatizirane Web usluge u nove aplikacije. SOA definira ove Web usluge, povezane komponente i infrastrukturu koja omogućava aplikacijama da budu sastavljene od usluga koje su napisane na različitim jezicima i koje su pokrenute na drugačijim ili distribuiranim platformama. SOA usluge surađuju prenošenjem poruka s jedne usluge na drugu ili usluga može sudjelovati u aplikaciji koordinirajući aktivnosti između drugih usluga [8].

Uz to, *Cloud Computing* se oslanja na *Web 2.0*, a to je inovativna tehnologija koja opisuje rastuće trendove koristeći *World Wide Web* tehnologiju i *Web* dizajn, a cilj je poboljšanje kreativnosti, dijeljenje informacija, kolaboracije i funkcionalnosti *Weba*. *Web 2.0* aplikacije tipično uključuju tehnike poput semantičkih *Web* tehnologija, spajanje sadržaja iz različitih resursa, alate za upravljanje korisnikovom privatnosti. Esencijalna ideja iza *Web 2.0* je napredak u povezanosti i aktivnosti *Web* aplikacija. Nova paradigma za razvoj i pristup *Web* aplikacijama omogućava korisnicima lakši i efikasniji pristup *Webu*. *Cloud Computing* usluge su po prirodi Web aplikacije koje učitavaju željene računalne usluge na zahtjev [9].

Tako da je *Cloud Computing* naslijedio mnoge prednosti i mane ovih tehnologija. Jedan od glavnih razloga razvijanja *Cloud Computinga* je da se u potpunosti iskoristi već postojeći, ali slabo iskorišteni računalni resursi u podatkovnim centrima [3].

Cloud computing je novonastala tehnologija od koje različite industrije i individue mogu imati velike koristi. On može biti iskorišten tako da pruža usluge koje bi se inače trebale instalirati na osobno računalo. Na primjer, pružatelji usluga mogu prodati uslugu korisnicima koja bi nudila prostor za pohranu korisničkih informacija ili usluga može dozvoliti korisnicima pristup i korištenje softverima koji su inače skupi [3].

Pojednostavljeno, *Cloud Computing* je troškovno efektivna računalna paradigma pri kojoj se informacije i procesna snaga može koristiti preko web pretraživača. On služi za pristup skalabilnim resursima poput podataka, datoteka, programa, hardvera. Korisnici plaćaju samo za korištene računalne resurse i usluge prema prilagođenom ugovoru o razini usluge – SLA (eng. *Service Level Agreement*) bez znanja o tome kako isporučitelj usluge koristi temeljnu računalnu infrastrukturu za potporu usluge. Servisno opterećenje u *Cloud Computingu* je dinamički promijenjeno prema zahtjevima krajnjih korisnika. Dakle, može se reći da se *Cloud Computing* još uvijek razvija te da ne postoji konačna definicija [9].

Također, novi rastući trend *Big Data*[1] svoju implementaciju čvrsto oslanja na postojanje cloud rješenja za podržavanje pohrane velike količine podataka kako bi se skalirala paralelna/raspoređena procesna snaga te kako bi se osigurao efektivan i siguran pristup mobilnih terminalnih uređaja heterogenim podatcima i uslugama. Očigledno, *Cloud Computing* evolucija je trenutno povezana s rastom popularnosti *Big Data*. Zapravo, *Cloud Computing* pruža potrebno računalstvo, pohranu, aplikacije i mrežu koji podržavaju *Big Data* aplikacije. Aplikacije podržane od strane *Cloud Computinga* mogu izvući vrlo korisne informacije kako bi se donosile bolje odluke u raznim područjima poput financija, industrije, transporta i slično [1].

2.3. Ugovor o razini usluge u okruženju *Cloud Computing*

Ugovor o razini usluge – SLA (eng. *Service Level Agreement*) je formalni sporazum dviju ili više stranki čime se određuju karakteristike usluge skupa s odgovornošću, prioritetima i pravima svih strana koje sudjeluju u sporazumu. Nadalje, određuje se cijena i model naplate

[1] Big Data računalstvo je rastuća znanstvena paradigma višedimenzionalnog rudarenja informacija za znanstvena otkrića i poslovne analize putem velike infrastrukture [51].

kazne u slučaju kada je ugovor prekršen, odnosno kada parametri isporučene usluge nisu u skladu s definiranim SLA ugovorom [10].

U okruženju *Cloud Computing*, jedna strana sporazuma je pružatelj usluge odnosno računalnog oblaka, a druga strana sporazuma je aplikacija smještena na infrastrukturu oblaka te ona predstavlja kupca. U ovom slučaju sporazum definira razinu kvalitete usluge pri kojoj aplikacija treba biti isporučena uzimajući u obzir iskorištenu infrastrukturu pružatelja usluge. Nasuprot, sama aplikacija u oblaku predstavlja uslugu koja se nudi krajnjim korisnicima, pa se ona definira posebnim sporazumom o razini usluge između aplikacije i njezinih korisnika [10].

Postoje dva tipa SLA – infrastrukturni i aplikacijski SLA. Infrastrukturni SLA je garantiranje dostupnosti i pouzdanosti hardverskih, odnosno infrastrukturnih resursa. On se prvenstveno odnosi na dostupnost napajanja, mreže, podatkovnih centara i slično. Infrastrukturni SLA sklapaju pružatelji računalnog oblaka s klijentima koji smještaju aplikacije na infrastrukturu oblaka. Ovim sporazumom se najčešće definira postotak vremena unutar kojeg pojedini resurs mora biti dostupan (npr. sklopovlje unutar podatkovnog centra treba biti dostupno 99,99% tijekom mjesec dana). Aplikacijski SLA definira performanse aplikacije i razinu usluge koja se pruža korisniku, a njega sklapa poslužitelj aplikacije s krajnjim korisnikom. Primjer stavke unutar aplikacijskog SLA je ograničenje vremena odziva web stranice (npr. uz broj istovremenih korisničkih konekcija manji od 1000, vrijeme odziva mora biti manje od jedne sekunde) [10].

3. Karakteristike koncepta *Cloud Computing*

Svojstva *Cloud Computinga* su ona koja pružaju velike količine moći u smislu računalstva i pohrane uz poboljšanu skalabilnost i elastičnost. Tako NIST definira pet temeljnih karakteristika *cloud computinga*:

- Samousluga na zahtjev
- Široki pristup mreži
- Udruživanje resursa
- Brza elastičnost
- Mjerena usluga [4].

3.1. Samousluga na zahtjev

Samousluga na zahtjev definira kako se korisnik može jednostrano opskrbiti sposobnostima kao što je vrijeme servera ili mrežna pohrana, po potrebi automatski bez zahtijevanja ljudske interakcije sa svakim pružateljem usluge [4].

Samousluga na zahtjev omogućava korisnicima upotrebu *cloud computing* resursa po potrebi bez ljudske interakcije između korisnika i pružatelja usluge. Korisnik može rezervirati korištenje *cloud* usluge kao što je računalstvo i pohrana po njegovim potreba uz dodatak upravljanja i razvijanja ovih usluga. Dakle, jednostavnost korištenja i eliminacija ljudskog faktora osigurava efikasnost i manje troškove i za korisnike i za pružatelje usluge [11].

3.2. Široki pristup mreži

Široki pristup na mreži je karakteristika koja omogućava pristup sposobnostima koje su dostupne preko mreže. Pristupa se preko standardnih mehanizama preko različitih klijentskih platformi (npr. mobilni uređaji, tableti i laptopi) [4].

Kako je *cloud computing* efikasna i efektivna zamjena za „kućni“ podatkovni centar, visoka propusnost komunikacijskih linkova mora biti dostupna za povezivanje *cloud* usluge. Mrežna povezanost s visokim prijenosnim pojasom pruža pristup velikoj količini resursa. Mnoge organizacije koriste arhitekturu u tri reda (funkcionalni procesi, podatkovna pohrana i korisničko sučelje su razvijeni i održavani kao nezavisni moduli na odvojenim platformama) ili arhitekturu u dva reda (sučelje je pokrenuto na klijentu, a podatkovni sloj je pohranjen na serveru) kako bi povezali različite računalne platforme poput laptopa i mobilnih uređaja na WAN (engl. *Wide Area Network*) [11].

3.3. Udruživanje resursa

Udruživanje resursa je sposobnost udruživanja isporučiteljevih računalnih resursa kako bi služili više korisnika koristeći dinamički dodijeljene i ponovno raspoređene različite fizičke i virtualne resurse prema korisničkim zahtjevima. Postoji prostorna neovisnost, odnosno korisnik generalno nema kontrolu ni znanje o točnoj lokaciji pruženih resursa te može specificirati lokaciju samo na višoj razini apstrakcije (npr. države). Primjeri resursa uključuju pohranu, procesiranje, memoriju i mrežnu propusnost [4].

Poslužiteljevi *cloud* računalni resursi se udružuju kako bi služili više korisnika koristeći „*multi-tenant*“ model. Aplikacije zahtijevaju resurse, ali ti resursi mogu biti geografski fizički locirani bilo gdje i dodijeljeni kao virtualne komponente kad god su potrebne [11].

3.4. Brza elastičnost

Brza elastičnost označava da se sposobnosti mogu elastično opskrbiti, u nekim slučajevima automatski zbog skaliranja izvana i iznutra proporcionalno zahtjevu. Za korisnika se često sposobnosti dostupne za opskrbu čine neograničene i mogu biti prisvojene u bilo kojoj kvantiteti u bilo koje vrijeme [4].

3.5. Mjerena usluga

Cloud sustavi automatski kontroliraju i optimiziraju korištenje resursa utjecanjem na sposobnost mjerenja na nekoj razini apstrakcije primjerenoj za tip usluge. Korištenje resursa može biti mjereno, kontrolirano i izvješteno, nudeći transparentnost za poslužitelja i korisnika usluge [4].

3.6. Tehničke karakteristike

Postoje dvije tehnički bitne tehničke karakteristike: labavo spajanje (eng. *loose coupling*) i tolerantnost na kvarove.

Labavo spajanje je tehnički fundament *cloud computinga*. Kroz virtualizaciju ili druge tehnologije, infrastrukture su odvojene u logičkom ili fizičkom smislu. Ponašanje jednog dijela teško utječe na ostale dijelove. Na primjer, platforma je apstraktni sloj koji može izolirati različite aplikacije pokrenute na njoj. Najvažnije od svega, cijeli *cloud computing* je pokrenut na klijent-server modelu. Klijenti ili *cloud* korisnici su labavo povezani sa serverima ili *cloud* pružateljima. Suprotan primjer je čvrsto spajanje sustava u kojem je naglasak na rješavanju znanstvenih problema [12].

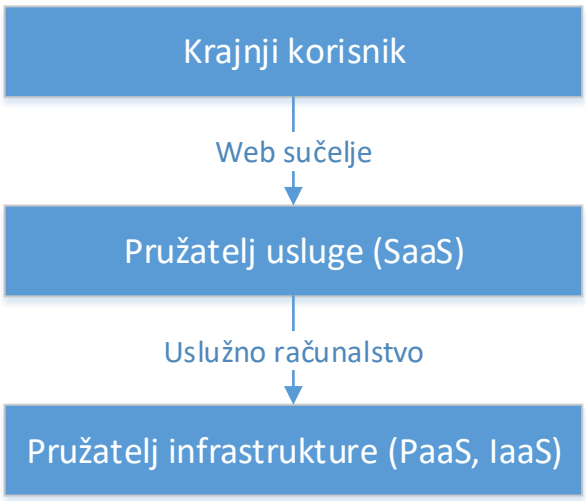
Postoji mnogo metoda za tolerantnost na kvarove u paralelnom računalstvu. Na niskoj razini, uvijek postoje neki korektivni mehanizmi sa specifičnim hardverom. Na visokoj razini, mnoge aplikacije su promatrate s metodama koje ciljaju na algoritme. Pri paralelnim računalnim sustavima na većoj skali, interval između dva kvara može biti kraći od vremena pokretanja aplikacije. Na primjer, neke znanstvene aplikacije su pokrenute tjednima i tijekom tog vremena se mogu dogoditi višestruki kvarovi. U takvim slučajevima tehnologija otporna na kvarove postaje od kritične važnosti. Nadalje, ne postoji ovisnost između različitih dijelova sustava, odnosno ne može doći do lančane reakcije [12].

4. Mogućnosti primjene koncepta *Cloud Computing*

Cloud Computing možemo podijeliti na više načina, prema modelima dostave te prema razvojnim modelima. Modeli dostave prema NIST-u su: server kao usluga, platforma kao usluga te infrastruktura kao usluga, a sve češće se spominju i podatci kao usluga. S druge strane, razvojni modeli dijele *Cloud Computing* prema vlasništvu, veličini i pristupu na: javni *cloud*, privatni *cloud*, *cloud* zajednica te hibridni *cloud*.

4.1. Modeli dostave *Cloud Computinga*

Cloud computing koristi uslužno orijentiran poslovni model. Drugim riječima, hardver i razine resursa su omogućene kao usluge po zahtjevu. *Cloud* nudi tri modela dostave: softver kao usluga – SaaS (eng. *Server as a Service*), platforma kao usluga – PaaS (eng. *Platform as a Service*) te infrastruktura kao usluga – IaaS (eng. *Infrastructure as a Service*) [13].



Slika 1. Model dostave Cloud Computinga

Izvor: [13]

Modeli dostave *Cloud Computinga* su prikazani na slici 1. Prema slojevitoj arhitekturi *Cloud Computinga*, sasvim je moguće da PaaS poslužitelj pokreće svoj *cloud* na IaaS poslužiteljevom *cloudu*. Ipak, u trenutačnoj praksi, IaaS i PaaS poslužitelji su najčešće dio iste organizacije (npr. Google i Salesforce) [13].

Tablica 1. Klasifikacija Cloud Computing usluga

Kategorija	Karakteristike	Oblik proizvoda	Pružatelji
SaaS	Korisnicima su pružene aplikacije pristupne od bilo kuda i bilo gdje	*Web* aplikacije i usluge (*Web 2.0*)	*Google documents, Google mail*
PaaS	Korisnicima su pružene platforme za razvoj aplikacija postavljenih na *Cloud*	Aplikacijsko programska sučelja i radni okviri; razvojni sustav	*Google AppEngine, Microsoft Azure*
IaaS	Korisnicima je pružen virtualni hardver i pohrana	Virtualne mašine, upravljanje infrastrukturom i pohranom	*Amazon EC2*

Izvor: [13]

Na tablici 1. su opisane glavne značajke usluga koje će u sljedećim odlomcima biti opisane detaljnije.

4.1.1. Softver kao usluga (SaaS)

Prema NIST-u, sposobnost koja je pružena korisniku je isporučiteljeva aplikacija pokrenuta na *cloud* infrastrukturi. Aplikacije su dostupne iz različitih klijentskih uređaja, preko web pretraživača ili preko programskog sučelja. Korisnik ne može upravljati ili kontrolirati osnovnu infrastrukturu *clouda* poput mreže, servera, operacijskih sustava, pohrane ili čak individualnih aplikacijskih sposobnosti s mogućom iznimkom ograničenih korisnikovih konfiguracija postavki aplikacije [4].

Softver ili aplikacija su usluga i pruženi su korisnicima preko Interneta. Ovaj model eliminira potrebu za instalacijom i pokretanjem aplikacije na korisničkom lokalnom računalu. Dakle, SaaS uklanja korisničku obavezu softverskog održavanja i smanjuje troškove kupovine softvera. SaaS rješenja su na vrhu *Cloud Computinga* stoga i oni pružaju krajnjim korisnicima integriranu uslugu kombinirajući hardver, razvojne platforme i aplikacije. Korisnicima nije dozvoljeno prilagođavati uslugu, ali dobivaju pristup specifičnoj aplikaciji postavljenoj na *Cloud*. Primjer SaaS implementacije su Google usluge poput Google Mail, Google Documents i Google Calendar koje su besplatno dostavljene Internet korisnicima. Primjeri komercijalnih rješenja su SalesForce.com koji nude online CRM (eng. *Customer Relationship Management*) i usluge projektnog upravljanja [9].

Prednosti SaaS-a su ta da održavanje i unaprjeđenja vrši poslužitelj te skalabilnost, a nedostatci su: minimalna prilagodba korisniku, integracija podataka, sigurnost i privatnost [11].

4.1.2. Platforma kao usluga (PaaS)

NIST definira PaaS na idući način: Sposobnost pružena korisniku je sposobnost razvitka korisnički razvijenih ili stečenih aplikacija stvorenih koristeći programske jezike, usluge i alate podržane od strane poslužitelja na *cloud* infrastrukturi. Korisnik ne može upravljati niti

kontrolirati temeljnu *cloud* infrastrukturu, ali ima kontrolu nad razvijenim aplikacijama i konfiguracijskim postavkama za aplikacijsko okruženje [4].

PaaS rješenja nude aplikacijske ili razvojne platforme u kojima korisnici mogu kreirati vlastite aplikacije koje će biti pokrenute na *Cloudu*. Preciznije, oni nude aplikacijske radne okvire i set aplikacijsko programskih sučelja – API (eng. *Application Programming Interface*)[2] koje mogu koristiti razvijači za stvaranje aplikacija za Cloud. PaaS rješenja često integriraju IT infrastrukturu na kojoj se mogu pokretati aplikacije. Ovakav je slučaj kod Google AppEngine i Microsoft Azure [9].

Google AppEngine je platforma za razvijanje skalabilnih web aplikacija koje su pokrenute na podatkovnim centrima koje održava Google. To definira aplikacijski model i pruža set API-ja koji omogućavaju razvijačima pristup dodatnim uslugama. AppEngine upravlja izvršavanjem aplikacija i automatski ih skalira po potrebi. Google pruža besplatnu, ali limitiranu uslugu dok iskorištava dnevne kvote za mjerenje i naplaćivanje aplikacija koje zahtijevaju profesionalnu uslugu [9].

Azure je Cloud usluga operacijski sustav koji služi kao razvojno i kontrolirano okruženje za Azure uslužnu platformu. Koristeći Microsoft Azure SDK, razvijači mogu stvoriti usluge koje iskorištavaju .NET radni okvir. Ove usluge moraju biti učitane kroz Microsoft Azure portal kako bi bile pokrenute na Windows Azureu. Dodatne usluge, kao što je upravljanje, orkestracija web usluga i pristup SQL podatkovnim spremištima, su pružene za izgradnju organizacijskih aplikacija [9].

Prednost PaaS-a je fokus na visokoj vrijednosti, a ne infrastrukturi te skalabilne sposobnosti. Rizik je model naplate te problemi s nadogradnjama [11].

4.1.3. Infrastruktura kao usluga (IaaS)

IaaS je usluga gdje je sposobnost pružena korisniku zapravo opskrba procesiranjem, pohranom, mrežama i ostalim osnovnim računalnim resursima gdje je korisnik u mogućnosti razvijati i pokretati proizvoljne softvere koji mogu uključivati aplikacije i operacijske sustave. Korisnik ne može kontrolirati i upravljati *cloud* infrastrukturom, ali ima kontrolu nad operacijskim sustavima, pohranom i razvijenim aplikacijama; te moguće ograničeno vrijeme kontrole određenih mrežnih komponenti (npr. domaćih vatrozida) [4].

IaaS je nastao kao rezultat ubrzanih napredaka u hardverskoj virtualizaciji, IT automatizaciji te korištenju mjerenja i naplate. IaaS rješenja dostavljaju IT infrastrukturu baziranu na virtualnim ili fizičkim resursima korisnicima. Ovi resursi susreću zahtjeve krajnjeg korisnika u obliku memorije, tipu centralne procesne jedinice i njene snage, pohrane te najčešće operacijskoj sustava. Korisnici postavljaju svoje aplikacije na ove resurse koji su upravljani u podatkovnim centrima [9].

Amazon je jedan od najvećih poslužitelja IaaS rješenja. Amazon Elastic Compute Cloud (EC2) pruža veliku računalnu infrastrukturu i uslugu temeljenu na hardverskoj virtualizaciji. Koristeći Amazon Web Services, korisnici mogu stvoriti Amazon Machine Images (AMIs) i

[2] Aplikacijsko programsko sučelje je set definicija, protokola i alata za izgradnju aplikacijskog softvera. Odnosno, set jasno definiranih metoda komunikacije između različitih softverskih dijelova [52].

spremiti ih kao predloške s kojih mogu biti pokrenute višestruke instance. Moguće je pokrenuti Windows ili Linux virtualne mašine za koje je korisnik terećen po satu svake pokrenute instance. Amazon također pruža usluge pohrane gdje korisnici mogu pohraniti veliku količinu podataka [9].

4.1.4. Podatci kao usluga (DaaS)

Podatci kao usluga – DaaS (eng. *Data as a Service*) je novonastali koncept koji nije definiran od strane NIST-a. DaaS je srodan softveru kao usluzi (SaaS) i temelji se na konceptu da se proizvod, odnosno podatak, može pružiti korisniku na zahtjev neovisno o geografskoj ili organizacijskoj odvojenosti pružatelja i korisnika [14].

Pristup podatcima u različitim oblicima i iz različitih resursa je omogućen uslugama na mreži. Na primjer, korisnici mogu manipulirati i brisati podatke kao što to rade na lokalnom disku ili pristupati podatcima u semantičkom smislu na Internetu. Amazon Simple Storage Service (S3) omogućava jednostavno Web sučelje koje se može koristiti za pohranu i povratak bilo koje količine podataka, u bilo koje vrijeme, s bilo kojeg mjesta na Webu [9].

4.2. Razvojni modeli *Cloud Computinga*

Postoji mnogo problema koje treba razmatrati pri premještanju organizacijskih aplikacija u *cloud* okruženje. Na primjer, neki pružatelji usluga su većinom zainteresirani u smanjenje cijene operacija, dok drugi više preferiraju visoku pouzdanost sigurnost [13]. Sukladno tome, NIST organizacija je definirala četiri vrste razvojnih modela *Cloud Computinga*: privatni *cloud*, *cloud* zajednica, javni *cloud* i hibridni *cloud*.

Unutar modela privatnog *clouda, cloud* infrastrukturu rukovodi samo jedna organizacija. S njom može upravljati i držati je u vlasništvu organizacija ili treća stranka [4]. Neki prodavači su koristili ovaj naziv za opis usluge emulacije *cloud computinga* na privatnu mrežu. Ove usluge su nudile mogućnost podržavanja aplikacija ili virtualnih mašina na tvrtkinim vlastitim poslužiteljima. Privatni *cloudovi* su privukli kritike jer ih korisnici i dalje trebaju kupovati, graditi i upravljati [11].

Kod modela *clouda* zajednice, *cloud* infrastruktura je podijeljena između zajednice korisnika iz većeg broja organizacija koji imaju zajedničke interese. Može biti u vlasništvu, upravljana i rukovođena od strane više organizacija u zajednici [4]. *Cloud* zajednice ima manje korisnika od javnog *clouda* i time se čini skuplji, ali može nuditi veću razinu sigurnosti i privatnosti. Primjer *cloud* zajednice je Google Gov Cloud [11].

Model javnog *clouda* odlikuje *cloud* infrastruktura koja je omogućena za otvoreno korištenje generalne javnosti [4]. Javni *cloud* pruža više ključnih koristi za pružatelja usluge, uključujući nedostatak inicijalnog ulaganja na infrastrukturu i prebacivanje rizika na pružatelje infrastrukture. Ipak, javni *cloudovi* nemaju dobru kontrolu nad podatcima te mrežnim i sigurnosnim postavkama što oštećuje njihovu efektivnost u mnogim poslovnim scenarijima [13].

Model hibridnog *clouda* označava *cloud* infrastrukturu koja je sastavljena od dvije ili više različite *cloud* infrastrukture koje ostaju zasebni entiteti, ali su vezani zajedno standardiziranom ili vlasničkom tehnologijom koja omogućava podatkovnu i aplikacijsku prenosivost [4].

Također, termin hibridni *cloud* može označavati i kombinaciju virtualnih *cloud* servera povezanih zajedno fizičkim hardverom. Prema [15], točna definicija hibridnog *clouda* bi bila korištenje fizičkog hardvera i virtualnih *cloud* servera zajedno kako bi se pružila jedna zajednička usluga. Hibridni *cloudovi* pohrane koriste kombinaciju javnog i privatnog *clouda* pohrane [11].

Virtualni privatni *cloud* (VPC) je alternativno rješenje za adresiranje ograničenja privatnog i javnog *clouda*. VPC je zapravo platforma pokrenuta na javnom *cloudu*. Glavna razlika je ta da VPC iskorištava tehnologiju virtualne privatne mreže – VPN (eng. *Virtual Private Network*) što omogućava pružateljima usluge dizajniranje vlastite tehnologije i postavki sigurnosti poput pravila vatrozida[3]. VPC je više holistički dizajn, jer ne samo da virtualizira servere i aplikacije, nego i temeljnu komunikacijsku mrežu. Uz to, za većinu tvrtki, VPC nudi besprijekorni prijelaz iz vlasničke infrastrukture u *cloud*-temeljenu infrastrukturu [13].

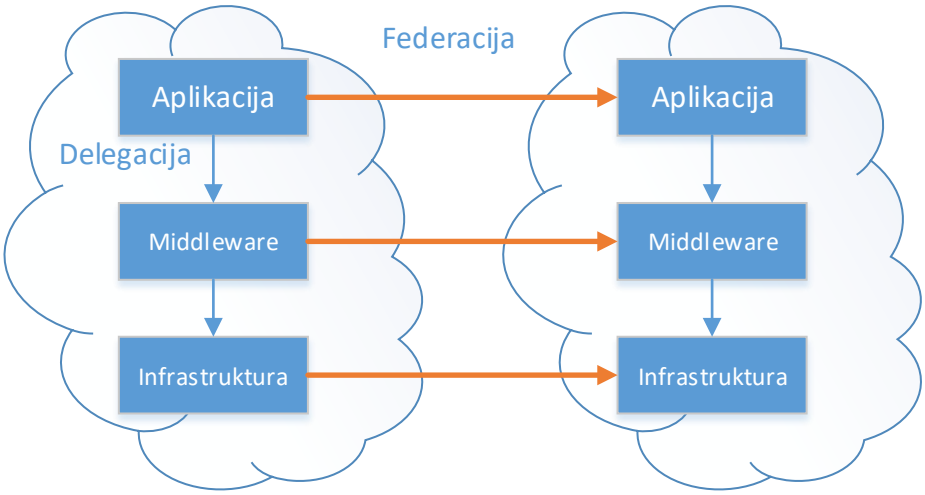
Za većinu pružatelja usluga, odabir pravog *cloud* modela je ovisno o poslovnom scenariju. Na primjer, računalno osjetljive znanstvene aplikacije su najbolje razvijene na javnom *cloudu* zbog troškovne efektivnosti [13].

4.3. *Cloud* federacija

Cloud federacija se sastoji od usluga iz različitih poslužitelja spojenih u jedno, podržavajući tri osnovne značajke međudjelovanja – migracija resursa, resursna redundancija i kombinacija komplementarnih resursa usluga. Migracija dozvoljava promjenu lokacije resursa poput slika virtualne mašine, podataka, izvornih kodova i sl. s jedne uslužne domene na drugu. Redundancija dozvoljava istovremeno korištenje sličnih uslužnih obilježja u različitim domenama, a kombinacija komplementarnih resursa i usluga dozvoljava kombinaciju različitih tipova u agregatne usluge [16].

Slika 2 prikazuje dva neovisna *Clouda* koji podržavaju vertikalni stog uslužnih slojeva od softverskog sloja (SaaS) na vrhu, kroz *middleware* ili platformski sloj (PaaS) do operacijskog sustava i infrastrukturnog sloja (IaaS). Na svakom sloju, izbor je napravljen da upotpuni uslužne zahtjeve kroz lokalne resurse [17].

[3] Vatrozid je mrežni uređaj sigurnosnog sustava koji nadgleda i kontrolira mrežni promet temeljem unaprijed definiranih sigurnosnih pravila [53].



Slika 2. Model *Cloud* federacije

Izvor: [17]

Na primjer, poslužitelj nudi SaaS uslugu privatnog ili javnog *Clouda*. Korisnik pošalje zahtjev aplikacijskom sloju koji procjenjuje je li dovoljno lokalnih resursa dostupno za zahtjeve usluge unutar određenog vremena. Ako aplikacijskih sloj ne može zadovoljiti ciljeve usluge, on može opcionalno zadovoljiti zahtjeve kroz nezavisni SaaS sloj poslužitelja iste usluge kao što je prikazano horizontalnom linijom između *Clouda* A i *Clouda* B. Rezultati se vraćaju korisnik kao da su lokalno stvoreni u aplikaciji pokrenutoj na *Cloudu* A. Federacija na SaaS sloju je analogna korištenju tradicionalnih „*peer*“ izvođača radova koji opskrbljuju ekvivalentne usluge primarnom poslužitelju olakšavajući elastičnost za dinamično tržište [17].

Drugi način bi bio da aplikacijski sloj poveća kapacitet kroz delegaciju. U delegaciji, aplikacijski sloj traži PaaS sloj u lokalnom *Cloudu* za dodatne resurse. Zahtjev za više resursa može biti ispunjen na više načina ovisno o dostupnosti trenutnog *Clouda*. Nedovoljni resursi se mogu zahtijevati od drugog *Clouda* u federaciji [17].

Iako je atraktivna iz poslovne perspektive, *Cloud* federacija zahtjeva nove tehnologije za efikasan rad. S obzirom da je slojeviti model, bitan dio je dizajn za izolaciju između slojeva. Na primjer, SaaS aplikacija dostavlja rezultate korisniku u određenom vremenu, ali da bi to postigla, ona ne treba poznavati detalje infrastrukture ispod nje [17].

4.4. Višekriterijska analiza *Cloud Computing* umrežavanja

Korisnici Cloud usluge trebaju odabrati prikladnu uslugu naspram svojih potreba. Kao jedna od rješenja, pojavila se višekriterijska analiza odabira – MCDA (eng. *multicriteria decision analysis*).

Komponente MCDA su: određen ili neodređen broj akcija, najmanje dva kriterija i jedan donositelj odluka. S ovim elementima, MCDA pomaže pri odlučivanju odabirom, sortiranjem i procjenjivanjem akcija. MCDA metode mogu biti podijeljene u dva tipa: više-atributna teorija korisnosti – MAUT (eng. *multiattribute utility theory*) i *outranking*. MAUT pokušava pronaći funkciju koja predstavlja korisnost određene alternative. Svakoj akciji je dodijeljena marginalna korisnost s stvarnim brojem koji predstavlja željenost promatrane akcije. Vraćena korisnost je

suma marginalnih korisnosti. *Outranking* metode određuju je li alternativa procijenjena više nego drugo preko usporedbe u parovima [18].

Analitički hijerarhijski proces – AHP (eng. *Analytic Hierarchy Process*) je najčešće korištena metoda za MCDA. AHP dozvoljava kvalitativnih i kvantitativnih kriterija pri evaluaciji alternativa, a atributi nisu u potpunosti neovisni jedni o drugih. AHP je baziran na usporedbi u parovima s atributima u hijerarhijskom odnosu. Hijerarhija započinje na najvišoj razini prema cilju; niže razine odgovaraju kriterijima, pod-kriterijima, itd. [18].

Pri korištenju metode AHP za dodjeljivanje resursa u *cloud computing* okruženju, zadatci su uspoređeni u parovima prema mrežnom frekvencijskom pojasu, vremenu završetka zadatka, cijene i pouzdanosti. Nakon toga, resursi su dodijeljeni. Nadalje, korištenjem AHP kako bi se obilježjima proizvoda odredio prioritet, dodijeljeni su SaaS proizvodi [18].

Iako je AHP efektivan alat za donošenje odluka, nedostatci poput subjektivnosti mogu iznijeti nesigurnosti pri određivanju usporedbi u paru. S toga je preporučen *fuzzy* AHP. *Fuzzy* označava set teorija koji koristi lingvističke vrijednosti predstavljene numeričkim varijablama. Ova metoda dozvoljava korištenje nejasnih odnosa na mjestu egzaktnih omjera. Unatoč tome, često je teško točno kvantificirati nečije mišljenje s brojem unutar intervala [0,1], pa se s tim intervalom radije označava stupanj uvjerenosti [18].

U mobilnom *Cloud Computingu*, jedan od ciljeva je odabir optimalnog *cloud* puta između određenih klasa *clouda* koje pružaju istu uslugu kako bi olakšali računalne zadatke. Kako je potrebno uzeti mnogo kriterija u obzir, poput brzine, frekvencijskog pojasa, cijene, sigurnosti, itd., višekriterijska analiza odlučivanja mora donijeti ispravne odluke. Ovdje se koristi AHP za određivanje težine kriterija odabira *cloud* puta, a *fuzzy* TOPSIS za obavljanje numeričke analize za rangiranje [18].

TOPSIS (eng. *Technique for Order of Preferences by Similarity to Ideal Solution*) je metoda koja prikazuje prednosti za sličnost s idealnim rješenjem koje pokušavaju pronaći alternativu koja je najbliža idealnom rješenju. U ovoj tehnici, matrica odlučivanja je prvo normalizirana, a nakon toga se idealna i ne-idealna rješenja identificiraju unutar normalizirane matrice [18].

TOPSIS je poznata MCDM metoda. Ona određuje najkraću udaljenost od pozitivnog idealnog rješenja i najdalju udaljenost od negativnog idealnog rješenja u cilju pronalaženja najbolje alternative. *Fuzzy* TOPSIS se koristio za odabir web usluge. Brojevi se koriste za predodžbu lingvističkih varijabli kao težinu kriterija i ocjenu web usluge [18].

5. Mrežna infrastruktura u konceptu *Cloud Computing*

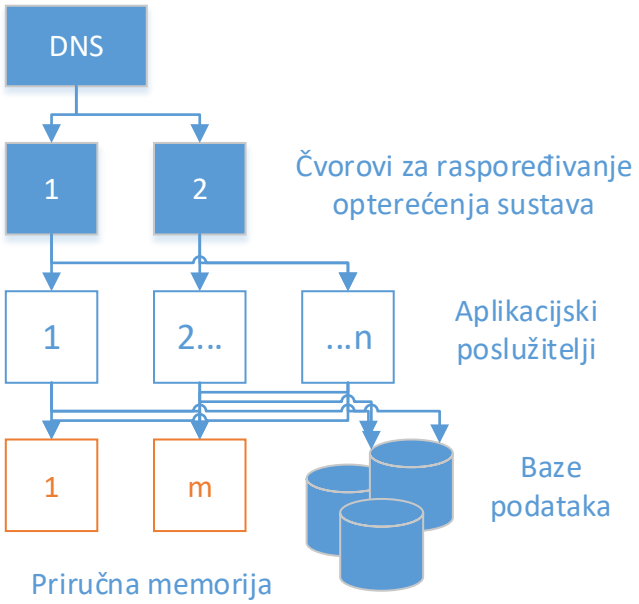
Cloud infrastruktura je osnova *cloud computinga* koja pruža mogućnosti računalstva, pohrane i mrežnih resursa, uključujući orkestraciju resursa, virtualizaciju i podjelu. Ona također omogućava slojevite funkcije koje podupiru gornji sloj usluga *clouda* [19].

5.1. Opći zahtjevi *cloud* infrastrukture

Cloud infrastruktura uključuje servere, pohranu, mreže i druge hardverske komponente. Ona može dostaviti infrastrukturne resurse kao uslugu. Virtualizacija omogućava podjelu jednog fizičkog dijela u više nezavisnih, samo-upravljajućih okruženja koji se mogu proširiti u smislu središnje procesne jedince, radne memorije, diska i drugih elemenata [19].

5.1.1. Skalabilnost

Cloud infrastruktura se mora brzo i fleksibilno prilagoditi promjenjivim zahtjevima korisnika i ostvariti visoku skalabilnost kroz različite mehanizme. Skalabilnost je sposobnost sustava da se prilagođava promjenama opterećenja uz jednaku razinu dostupnosti i pouzdanosti [20]. Postoje dvije vrste skalabilnosti, horizontalna i vertikalna. Horizontalna skalabilnost je povećanje broja čvorova odnosno računalnih resursa u sustavu, a vertikalna skalabilnost se odnosi na poboljšanje dijelova. Skalabilnost infrastrukture označava skalabilnost poslužitelja i mrežnih resursa [10].



Slika 3. Referentna arhitektura *Cloud Computinga* za skalabilnost

Izvor: [10]

Na slici 3 prikazana je referentna arhitektura računalnog oblaka s obzirom na najbolju praksu vezanu uz skalabilnost infrastrukture sustava. Kao što se vidi na slici, infrastruktura je podijeljena u slojeve. Prvi sloj nakon poslužitelja domenskog imena sustava - DNS (eng.

Domain Name System)[4] se sastoji od čvorova za raspoređivanje opterećenja sustava. Nakon toga slijedi sloj aplikacijskih poslužitelja i sloj priručne memorije između aplikacijskih poslužitelja i baze podataka [10].

Da bi se povećala skalabilnost u referentnoj arhitekturi, najjednostavnije je povećati broj aplikacijskih poslužitelja. S druge strane, u slučaju očekivanog velikog opterećenja, predlaže se horizontalna skalabilnost čvorova za raspoređivanje opterećenja. Skalabilnost aplikacija se također može poboljšati prikladnom konfiguracijom sloja priručne memorije (eng. *cache*). Pored priručne memorije, skalabilnost baze podataka je neophodna za ostvarenje skalabilnosti platforme. Naime, riječ je o horizontalnoj skalabilnosti gdje se stvara više instanci sustava za upravljanje baze podataka, ali se s njihovim povećanjem, povećava i složenost održavanja stanja svih replika [10].

Problem skalabilnosti u oblaku je osiguranje pravovremenog zauzimanja novih i pravovremeno otpuštanje nepotrebnih resursa. S toga je većina istraživanja usmjerena prema dinamičkoj skalabilnosti. Dinamička skalabilnost je automatsko uočavanje nedostatka resursa tijekom izvođenja aplikacije, detekcija nepotrebnih resursa i zauzimanje novih resursa. Dinamička skalabilnost koja se odnosi na jednog pružatelja računalnog oblaka dovodi do mogućnosti dijeljenja resursa između više različitih pružatelja infrastrukture. Primjer takvog djelovanja je *InterCloud* [10].

5.1.2. Konvergencija

Konvergencija označava zahtjev da *cloud* infrastruktura prenese upravljački promet, backup promet i promet pohrane iz reda centralizirane pohrane do servera na istoj mreži koja prenosi IP podatke. Općenito, koncept konvergencije je „jedna žica“ (eng. *one wire*) što omogućava fleksibilnu U/I infrastrukturu s znatno smanjenim zahtjevima za kabelom. Primjer ovakve tehnologije je *fibre channel* preko *Etherneta* – FCoE (eng. *Fibre Channel over Ethernet*) [19].

5.1.3. Resursno usklađivanje

Mrežni resursi trebaju biti usklađeni s IT resursima, što znači da mrežna oprema treba poznavati status i konfiguraciju IT resursa. Usklađivač zaprima zahtjeve za *cloud* usluge od korisnika i predaje ove zahtjeve različitim dijelovima uključenim u slaganje i dostavu cloud usluge. Različite komponente uključuju: operacijski sustav podrške – OSS (eng. *operational supporting system*)[5] podatkovnog centra *clouda*, OSS prijenosne mreže i mogući OSS *inter-cloud* mreže [19].

5.1.4. Performans

Priroda prometa u *cloudu* je poprilično nasumična. *Cloud* poslužitelji moraju transformirati lokalne mreže podatkovnih centara u lokalne mreže distribuiranih sustava s pouzdanom i brzom *point-to-point* komunikacijom između servera s linkovima bez zagušenja i jednoličnom propusnošću između bilo koja dva servera unutar podatkovnog centra. Iz

[4]DNS je distribuirana baza podataka namijenjena za hijerarhijsko imenovanje svih terminalnih uređaja spojenih na Internet ili mrežu [54].

[5] OSS je računalni sustav koji koriste pružatelji Internet usluga, a namijenjen je održavanju mrežnih elemenata, rezerviranju kapaciteta, konfiguraciji mrežnih elemenata i upravljanju greškama [55].

perspektive arhitekture, trenutačna topologija sastavljena od pristupa, agregacije i jezgre, korištena u podatkovnim centrima nije dobro prilagođena za pružanje ovih zahtjeva. Druga stvar za razmatranje je kako maksimizirati korištenje dostupnih veza [21].

5.1.5. Dinamična migracija virtualne mašine

Dinamična migracija virtualne mašine označava prijenos operacijskog sustava i njegovih aplikacija s virtualne mašine na drugu virtualnu mašinu bez zaustavljanja rada operacijskog sustava. Dinamična migracija se također naziva živa migracija. Dinamična migracija omogućava operaciji virtualne mašine visoku dostupnost, dinamičnu distribuciju tereta i tijek usluge tijekom održavanja hardvera [19].

Tijekom dinamične migracije, IP (eng. *Internet Protocol*)[6] adresa virtualne mašine bi trebala ostati nepromijenjena ako je to moguće (npr. migracija unutar iste mreže u mrežnom sloju). Sve police poput kvalitete usluge i sigurnosti za virtualnu mašinu na mrežnoj opremi bi se trebali migrirati u isto vrijeme, tako da se osigura kontinuitet, pouzdanost i sigurnost [19].

Virtualna mašina se mora moći migrirati između fizičkih servera. Tijekom tog procesa, gostujući operacijski sustav ne mora znati za migraciju. Virtualno okruženje uključujući memoriju, pohranu i mrežnu konfiguraciju je zadržano isto kao i prije. Ipak, neizbježan je kratak interval kad se finalne dinamične informacije (npr. vrijednosti registra virtualne centralne procesne jedinice) prenose na novu virtualnu mašinu [19].

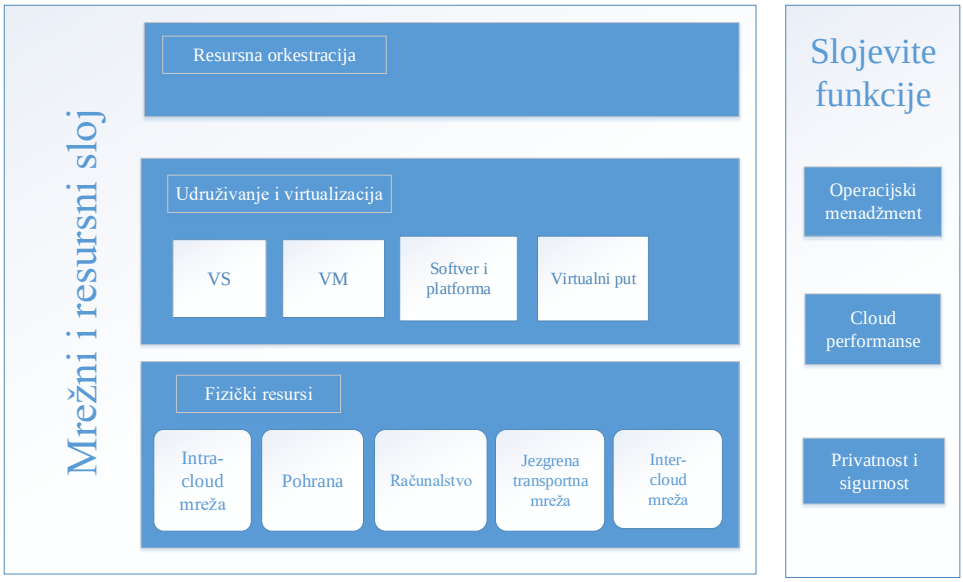
Jednom kad *cloud* sustav završi svoj proces, virtualna mašina na ciljanom poslužitelju će početi raditi, a virtualna mašina na izvornom poslužitelju će završiti s radom. Tijekom ovog procesa, uređaji drugog sloja mreže (npr. prospojnik) moraju znati za ovu promjenu. Također, ako je migracija na poslužitelja s drugačijim IP *subnetom*, potrebno je ažurirati treći sloj mreže i promijenit će se IP adresa virtualne mašine [19].

5.2. Radni okvir *cloud* infrastrukture

Cloud infrastruktura je esencijalna komponenta *cloud* arhitekture. Ona omogućava računalne i mrežne mogućnosti te mogućnosti pohrane skupa s relevantnim slojevitim potpornim funkcijama za podupiranje *cloud* usluge.

Slika 4 prikazuje radni okvir *cloud* infrastrukture. Logično, *cloud* infrastruktura uključuje sloj resursa i sloj funkcija. Resursni sloj je podijeljen u tri pod-sloja: resursnu orkestraciju, udruživanje i virtualizacija te fizičke resurse. Funkcijski sloj sadrži: sigurnost i privatnost, *cloud* operacijski menadžment i *cloud* funkcije performansi [19].

[6] Internet protokol je standardni internetski protokol čije su osnovne funkcije adresiranje i usmjeravanje, a IP adresa je jedinstvena brojčana oznaka duga 32 bita koja označava svako računalo na Internetu [54].



Slika 4. Radni okvir za *cloud* infrastrukturu

Izvor: [19]

Resursno upravljanje je definirano kao kanaliziranje upravljanja, praćenja i zakazivanja računalstva, pohrane i mrežnih resursa u iskoristive usluge za više slojeve i korisnike. Resursno upravljanje kontrolira stvaranje, modifikacije, prilagodbu i uklanjanje virtualnih resursa. Ono održava direktorije o tome što je moguće unutar *cloud* segmenta temeljem ukupnog kapaciteta resursa i inkrementalnih namjena.

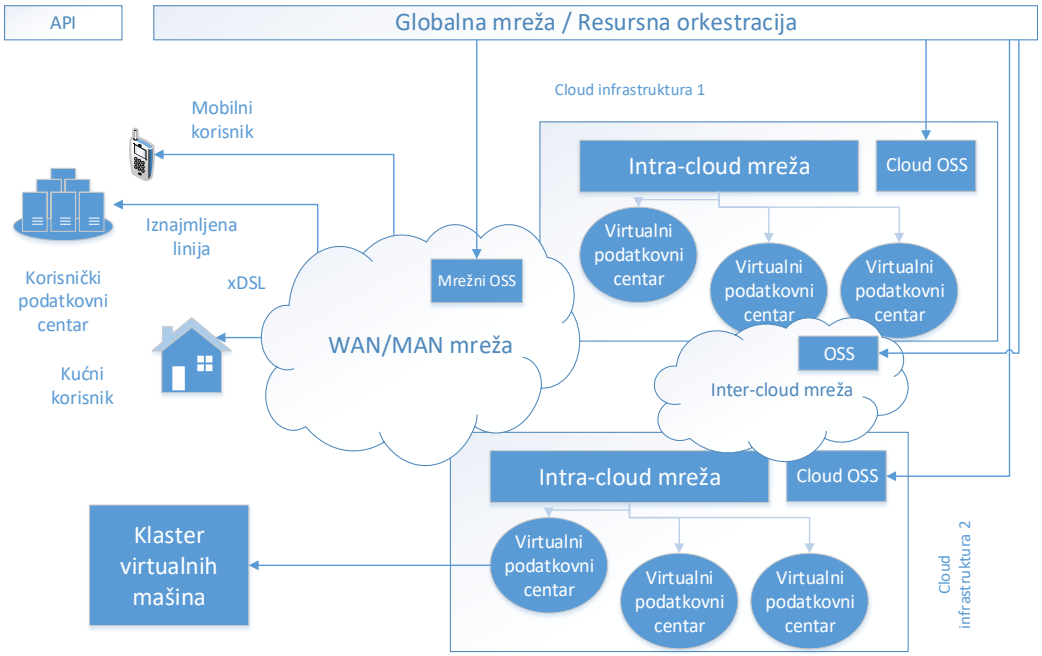
Udruživanje i virtualizacija fizičkih resursa su temelj pri dostizanju na-zahtjev i elastičnih karakteristika *cloud computinga*. Kroz ove procese, fizički resursi su pretvoreni u virtualne mašine, virtualnu pohranu i virtualne mreže. Ovi virtualni resursi su zauzvrat upravljani resursnom orkestracijom temeljem korisničkih zahtjeva [19].

Fizički resursi se odnose na resurse koji su temeljni za ostvarivanje *cloud* usluge. Ovi resursi uključuju sve što se nalazi unutar *cloud* podatkovnih centara (npr. računalni serveri) i sve što se nalazi van podatkovnog centra, tipično mrežni resursi.

Slojevite funkcije uključuju sigurnost i privatnost, *cloud* operacijski menadžment i *cloud* funkcije performansi. Ovaj sloj zadovoljava pružateljevu potrebu za praćenjem resursa i generira ujedinjen pogled na trenutnu namjenu resursa i kako efikasno iskoristiti resurse. Resursno praćenje dozvoljava pružatelju usluge izvođenje balansiranje opterećenja kako bi se osigurao performans *cloud* usluge. Ono također označava probleme poput hardverskih i softverskih kvarova ili sigurnosne probleme [19].

5.3. Mrežni model *cloud* infrastrukture

Postoji mnogo mrežnih komponenti uključenih u isporuku i stvaranje *cloud computing* usluge, a predstavljene su na slici 5.



Slika 5. Mrežni model *cloud computinga*

Izvor: [19]

Intra-cloud mreža je mreža koja povezuje lokalne *cloud* infrastrukture kao što je LAN podatkovni centar korišten za povezivanje servera. Ona obavlja tri tipa prometa:

- Mreža između virtualnih mašina na istom serveru
- Mreža između servera i sustava pohrane (npr. *Fiber Channel over Ethernet* je tipična implementacija)
- Mreža između virtualnih mašina na različitim serverima

Jezgrena transportna mreža (WAN/MAN) je mreža koju koriste korisnici kako bi pristupili uslugama razvijenim unutar poslužiteljevog podatkovnog centra, a *inter-cloud* mreža služi za povezivanje *cloud* infrastruktura zajedno koje mogu imati različite ili iste vlasnike. Ove tri mrežne komponente su osnova svih *cloud* usluga i njihovog dostavljanja [19].

Operacijski sustav podrške u *cloud computingu* je derivacija OSS telekomunikacijske mreže, a postoje dvije kategorije: mrežni OSS i *cloud* OSS. Mrežni OSS je tradicionalni OSS u sustavu dodijeljen poslužiteljima telekomunikacijskih usluga, a *cloud* OSS je sustav dodijeljen pružateljima *cloud computing* usluga kako bi podupirao procese poput održavanja i konfiguracije *cloud* resursa [19].

5.3.1. Funkcionalni zahtjevi za intra-cloud mrežu

Tradicionalna hijerarhija prospojnika (eng. *switch*)[7] je stablasta topologija unutar podatkovnog centra koji se sastoji od tri sloja: pristup, agregacija i jezgra. Pristupni sloj i agregacijski sloj su povezani podatkovnim slojem kojeg pokreće STP (eng. *spanning tree protocol*). STP protokol je mrežni protokol na drugom sloju koji kreira logičku topologiju bez

[7] Prospojnik je uređaj s više priključnica koji prima okvir podatkovne razine i prosljeđuje ih na odredište [53].

petlje za *Ethernet* mreže. Ova shema garantira da postoji samo jedan put između bilo koje dvije točke (servera) i da neće biti petlji s blokiranim linkovima. U slučaju kvara linka, mreža zaustavlja sav promet dok STP algoritam kalkulira dozvoljene puteve bez petlje kroz mrežu. Za tradicionalne podatkovne centre, ovakva STP – temeljena mrežna topologija radi [22].

Unutar virtualne *cloud* infrastrukture, svaki server može postati virtualan s mnogo virtualnih mašina i svaka virtualna mašina se ponaša kao jedan server te ima iste konekcijske zahtjeve što stvara sve više horizontalnog prometa između virtualnih mašina unutar podatkovnog centra. Ipak, mrežna topologija temeljena na STP-u održava samo jedan put između ovih servera i ne može zadovoljiti ove zahtjeve. Nadalje, tradicionalne MAC (eng. *Media Access Control*)[8] adrese nisu slojevite što dovodi do opterećenog adresnog prostora, naročito u virtualnim podatkovnim centrima [19].

5.3.2. Funkcionalni zahtjevi *inter-cloud* mreže

S razvojem *cloud*-usmjerenih usluga, raste i količina prometa između podatkovnih centara. Sav promet između podatkovnih centara mora proći kroz osnovnu mrežu što je znatno optereti. Prometni modeli su različiti s različitim aplikacijama, poput podatkovne sinkronizacije ili velike podatkovne replikacije. Uz to, različiti podatkovni centri imaju različite zadatke (transakcijski, procesni, pohrana). Oni zahtijevaju različite prometne modele koji će garantirati različite zahtjeve latencije, osjetljivosti na gubitak paketa i propusnost [19].

Inter-mrežna topologija budućih podatkovnih centara treba zadovoljiti veći stupanj agilnost. Kako bi se smanjilo opterećenje osnovne mreže, potrebno je razlikovati promet koji može putovati kroz osnovnu mrežu i promet koji se može prenositi dodijeljenom mrežom između podatkovnih centara. Uz ovo, paketi bi se usmjeravali efektivno i efikasno, jer se može ostvariti interakcija s mrežom u smislu da se topologije prilagode specifičnim zahtjevima [19].

U mrežnoj topologiji, izlazni usmjeritelj *cloud* podatkovnog centra je povezan s agregacijskim usmjeriteljem MAN (eng. *Metropolitan Area Network*) mrežom. S obzirom da različiti *cloud* podatkovni centri sadrže infrastrukturu i uređaje koji pripadaju istom *cloud* pružatelju usluga, ovi *cloud* podatkovni centri mogu biti povezani posebnom linijom ili direktno preko *fibre channel* što bi se smatralo djelom intra-mreže u *cloud* infrastrukturi poslužitelja [19].

5.4. Pohrana

Sloj pohrane je sloj gdje informacija putuje i „živi“. Bez njega, u *cloud computingu* ništa se ne bi dogodilo. Ovisno o dizajnu *clouda,* pohrana može biti bilo koje rješenje, a neki od najpoznatijih su obrađeni ispod.

5.4.1. RAID

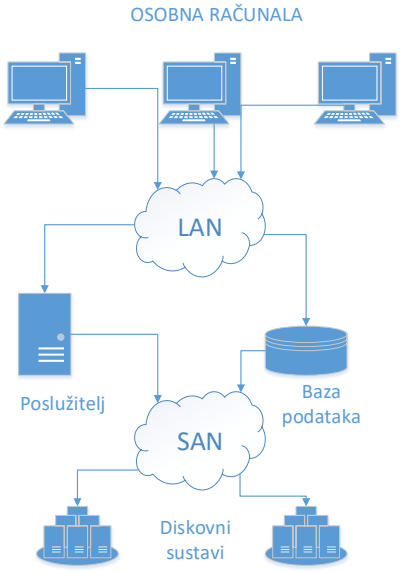
Redundantni spremnici za pohranu – RAID (eng. *Redundant Array of Independent Disks*) je koncept koji omogućava jednostavnije kopiranje podataka s pouzdanim, na kvar tolerantnim sustavom. RAID kontrolor može održavati diskove u sinkronizaciji i upravljati svim upisima i učitavanjima ulaza/izlaza sa i od diskova. Najpoznatija vrsta RAID-a je RAID 5. On se sastoji

[8] MAC adresa računala je jedinstveni identifikator dodijeljen mrežnim sučeljima za komunikacijama na podatkovnom sloju mreže [53].

od podjele korisničkih podataka u N-1 dijelova gdje je N boj diskova koji se koriste za izgradnju RAID sustava. RAID 5 zahtjeva minimalno tri diska. U teoriji, ne postoji ograničenje pri dodavanju diskova. Hibridni RAID sustavi koriste kombinaciju postojećih RAID razina kako bi kreirali RAID s povlasticama dvaju RAID-a odvojenih razina [23].

5.4.2. SAN

SAN (eng. *storage area network*) je mreža čija je primarna svrha prijenos podataka između računalnih sustava i elemenata pohrane. SAN se sastoji od komunikacijske infrastrukture što označava fizičku povezanost. On također uključuje sloj menadžmenta koji organizira povezanost, elemente pohrane i računalne sustave tako da je prijenos podataka siguran i robustan. Jednostavnije, SAN je specijalizirana, brza mreža koja povezuje servere i uređaje pohrane. SAN se nekad naziva kao „mreža iza servera“[23]. Ona omogućuje povezivanje udaljenih uređaja za pohranu na poslužiteljima tako da izgledaju kao da su lokalno spojeni na operacijski sustav. Slika 6 prikazuje tipičnu SAN mrežu koja je najčešće na odvojenoj mreži od LAN mreže kako ne bi utjecala na njen promet, a elementi spojeni na ovoj slici su spojeni kao čvorovi (eng. *peers*) [24].



Slika 6. SAN mreža

Izvor: [24]

Arhitektura potrebna za podršku SAN mreže je jako slična arhitekturi DAS sustava. Sastoji se od aplikacije, operacijskog sustava te uređaja za prijenos. Osnovna razlika naspram DAS sustava je u tome što je diskovni kontrolor zamijenjen sa FC ili iSCSI/TCP/IP slojem koji osiguravaju funkcije transporta do udaljenog diskovnog sustava [24].

Iako je moguće spojiti LAN i SAN infrastrukturu, podjela ipak daje više prednosti. Na primjer, SAN mreža može zahtijevati veliki kapacitet pri stvaranju sigurnosnih kopija na određeni vremenski period što LAN mreža ne može osigurati [24].

5.4.3. NAS

Umrežena spremišta podataka ili NAS (eng. *Network-attached storage*) tehnologije su samostalna računala priključena na mrežu s namjenom da pružaju uslugu pohrane podataka drugim uređajima u mreži.

Operacijski sustav i programi NAS tehnologije omogućuju pohranu podataka i pristup datotekama. Najčešće, web preglednik se spaja na mrežnu adresu računala. NAS sustavi sadrže jedan ili više tvrdih diskova, često uređene u logičke, redundantne spremnike za pohranu ili RAID nizove. Naime, ovakva tehnologija omogućava arhiviranje podataka na jeftine diskovne komponente preko nizanja uređaja u redundantne liste. Redundancija se ostvaruje kreiranjem sigurnosnih kopija podataka (eng. *mirroring*), zapisivanjem podataka preko liste za ispravljanje pogrešaka (eng. *parity data*) ili spremanjem podataka preko više diskova (eng. *striped*) [24].

Prednost NAS tehnologije je jednostavnost uporabe i jednostavna implementacija u LAN okruženjima bez potrebe za novom mrežom infrastrukturom. Nadalje, omoguće je implementirati heterogenu mrežu s pristupom različitim operacijskim sustavima. Operacijski sustavi poput Linuxa i Unixa, uključuju podršku za NAS protokole, a novije inačice Windowsa podržavaju CIFS protokol. Jedno od najvažnijih obilježja NAS tehnologije je sigurnost podataka uz pomoć RAID lista, a odvajanje spremišta podataka s računala na odvojeno mjesto povećava performanse sustava [24].

S druge strane, NAS sustav ima ograničene resurse što predstavlja ograničen broj korisnika i zahtjeva. NAS uređaj je ograničen na vlastito sklopovlje, pa ga najčešće nije moguće nadograditi [24].

5.4.4. Hibridni sustav NAS i SAN

Povezano rješenje, to jest hibridni sustav NAS i SAN tehnologije omogućava prilagodljivost i poboljšanje performansi za većinu organizacija. NAS sustav je važan za heterogena okruženja, a SAN za okruženja u kojima se razmjenjuje puno podataka. Posjedovanjem NAS sustava omogućuje se lakši pristup SAN mreži. Zapravo, NAS sustav je idealan poveznik (eng. *gateway*) prema SAN mreži koji usmjerava podatke iz SAN mreže do odgovarajućih poslužitelja u obliku datoteka. Nadalje, SAN mreža omogućava učinkovitiji rad NAS tehnologije jer smanjuje opterećenje pohrane manje kritičnih podataka. Važne podatke je moguće pohraniti lokalno na NAS uređaju dok se manje važni podatci mogu prenijeti SAN mreži [24].

5.4.5. DAS

DAS (eng. *Direct Attached Storage*) označava povezivanje tvrdih diskova ili niza na osobno računalo čime nastaju ograničenja u pristupu i rukovanju podatcima. Klasična DAS konfiguracija sadrži računalo koje je izravno povezan na jedan ili nekoliko tvrdih diskova. Između diskova se najčešće koriste standardi poput SCSI ili FC. Ako se koriste sustavi za pohranu umjesto diskova, oni najčešće sadrže RAID liste ili sustave hrpe diskova - JBOD (eng. *Just Bunch Of* Disks). Radi se o skupini tvrdih diskova kojima se pristupa kao odvojenim uređajima. Postoje brojni načini povezivanja diskova u JBOD sustav, a ovise i o tipu tvrdog diska [24].

Glavno obilježje DAS sustava je povezivanje resursa za pohranu s jednim računalom ili poslužiteljem. Međutim djelotvornost resursa za pohranu je niska jer je sav kapacitet vezan za jedno računalo. Kapacitet DAS sustava je ograničen na broj diskova, a dodavanje ili uklanjanje diska može narušiti pristup ostalim diskovima [24].

5.5. Poslužitelji

Svaka serverska platforma (IBM z systems, UNIX, Linux, Microsoft Windows Servers) implementiraju SAN rješenja koristeći različite interkonekcije i tehnologije pohrane. Iduće sekcije pregledavaju ova rješenja i implementacije na svakoj platformi [23].

5.5.1. *Mainframe* poslužitelj

Mainframe je jedinstven računalni sustav s potencijalno više procesora te visokim performansama. Iako evolucija tehnologije ide prema distribuiranoj infrastrukturi, *mainframe* računala još uvijek igraju bitnu ulogu u poslovanjima koja se temelje na masivnim mogućnostima pohrane.

IBM System z je procesor i operacijski sustav u *mainframe* setu. IBM z Systems podržava različite operacijske sustave, a veza između procesora i uređaja pohrane se razvila iz sabirnica u FICON (eng. *Fibre Connection*) kanale. IBM ima standardizirane *Fibre Channel* adaptere koji se koriste s z Systems serverima koji može implementirati Linux [23].

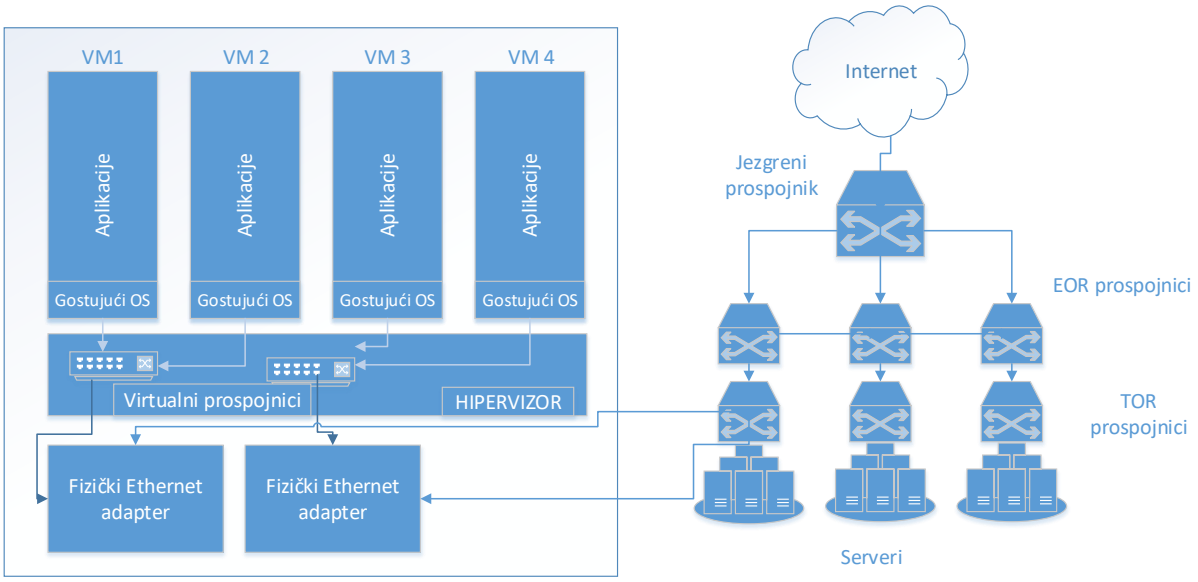
5.5.2. UNIX poslužitelji

UNIX poslužitelji su originalno dizajnirani za računalne sustave s visokim performansama poput *mainframea*, a današnji UNIX operacijski sustav se pojavljuje u mnogim hardverskim platformama, od Linux osobnih računala do dodijeljenih radnih stanica. Različiti prodavači UNIX sustava na tržištu razvijaju različite varijacije UNIX operacijskog sustava gdje svaki proizvod nudi jedinstvena unaprjeđenja. Tako različite implementacije često podržavaju različite podatkovne sustave [23].

5.6. Mrežni sloj *Cloud Computinga*

Iako se *cloud comuting* nužno ne oslanja na virtualizaciju, većina *cloud* infrastrukture je izgrađena uz pomoć virtualnih servera. Unutar virtualnog okruženja, neke se mrežne funkcionalnosti poput prospajanja i vatrozida nalaze unutar fizičkog servera [22]. Mrežna arhitektura podatkovnih centara u *cloud*-temeljenom okruženju se sastoji od tri glavne razine: jezgra, agregacija i pristup.

Jezgra se sastoji od jezgrenih prospojnika povezanih na Internet ili WAN preko IP sloja ostvarujući konačnu agregacijsku točku za mrežu i upravljanje agregacijskim prospojnicima kako bi radili zajedno. Agregacija uključuje agregacijske prospojnike, također poznati pod EOR (eng. *End-Of-Rack*) prospojnike koji služe za pristup drugim serverima stvaranjem prikladnih veza između sebe i pružajući veliku propusnost [25]. EOR prospajanje implicira postavljanje prospojnika u dodijeljena postolja za agregaciju prometa iz različitih postolja i povezivanje s jezgrenom mrežom [26]. Pristupni sloj je sloj gdje je veza između servera i EOR prospojnika organizirana preko TOR (eng. *Top-of-Rack*) *Ethernet* prospojnicima [25]. TOR prospojnici skupljaju promet s jednog postolja i nude povezanost jezgrenoj mreži. TOR zahtjeva više prospojnika od EOR tehnike, ali pojednostavljuje postavljanje kablova. Uz to, on je jedina danas dostupna tehnologija za rješenja koja podržavaju i *Ethernet* i *Fiber Channel* [25].



Slika 7. Tipična mrežna arhitektura prospojnika podatkovnog centra

Izvor: [22]

Uzmimo u obzir softversko-orijentiranog virtualnog prospojnika prikazanog na slici 7. Virtualni prospojnik u istom fizičkom serveru se može koristiti za prospajanje prometa između virtualnih mašina i skupljanje prometa za konekciju prema eksternom fizičkom prospojniku. Virtualna mašina je često implementirana kao priključak hipervizoru. Virtualne mašine imaju virtualne *Ethernet* adaptere koji se povezuju s eksternim *Ethernet* prospojnikom. Za razliku od fizičkih prospojnika, virtualni prospojnik ne mora nužno pokretati mrežne protokole kako bi funkcionirao, niti se treba odnositi prema svim portovima jednako jer zna da su neki od njih povezani na virtualne *Ethernet* portove. On funkcionira uz prikladnu konfiguraciju od strane vanjskog menadžera [22].

Mrežna arhitektura je jedna od ključnih blokova *cloud computinga*. *Cloud* korisnik se spaja na mrežu kako bi pristupio mrežnim resursima. *Cloud* je pristupačan preko javne mreže (Interneta) ili privatne mrežne infrastrukture (npr. dodijeljeni link). Najvažniji efekt *cloud computinga* na mrežu su podatkovni centri. Podatkovni centri se sastoje od servera međusobno povezanih preko TOR *Ethernet* prospojnika koji prenosi konekcije prema agregacijskom, EOR prospojniku. EOR prospojnik se povezuje s drugim EOR prospojnicima i preko njih na druge servere u podatkovnom centru. Jezgreni prospojnik je povezan različitim EOR prospojnicima i omogućava vezu s vanjskim svijetom, tipično kroz treći sloj OSI RM (IP). S obzirom da većina prometa unutar podatkovnog centra prelazi preko TOR i EOR prospojnici, stablasta topologija najbolje prikazuje taj slučaj, a prisutnost virtualnih servera dodaje dodatnu dimenziju. U usporedbi s mrežnom povezanosti za običan server, mrežna povezanost s fizičkim serverima se treba povećati jer je promet za više virtualnih mašina multipleksiran na istu fizičku *Ethernet* konekciju [22].

IaaS je sloj koji pruža fleksibilnost, odnosno korisnici su naplaćeni prema korištenju računalnih operacija, pohrane itd. Ipak, bitan je faktor prijenos podataka naprijed i nazad između *cloud* korisnika i *cloud* pružatelja usluge. Većina IaaS poslužitelja naplaćuje količinu podataka prenesenu preko linka. Ove naplate se brzo nagomilaju ako aplikacije zahtijevaju

mnogo podatkovnog prometa. Još jedna briga vezana za ovaj aspekt je vrijeme koje može potrošiti inicijalni *upload* ili *download* [22].

Neke aplikacije i protokoli mogu imati koristi od uređaja za WAN akceleraciju koji se mogu postaviti na oba kraja WAN *linka*. Ovi uređaji nisu specifični za *cloud* – oni se koriste već godinama za poboljšanje performansi kad je u pitanju WAN link [22].

VM migracija donosi sasvim drugi set problema. Najpoznatiji scenarij je kad se VM migrira na drugog poslužitelja unutar iste topologije u sloju 2 (uz pripadajući virtualni LAN, VLAN i konfiguraciju). Na primjer, migracija VM s otvorenim TCP[9] konekcijama. Ako se koristi dinamična migracija, TCP konekcije neće osjetiti nikakve promjene, ali nakon migracije, IP i TCP paketi određeni za VM će se rješavati s drugačijom MAC adresom ili istom MAC adresom sada povezanom na drugi fizički prospojnik u mreži. Predložena rješenja uključuju netražene ARP (eng. *Address Resolution Protocol*) zahtjeve od migrirane VM tako da prospojne tablice mogu biti ažurirane ili pseudo MAC adresa za VM kojom se upravlja izvana. Alternativno, VM može biti migrirana preko drugog ili trećeg sloja mreže s raskidanjem TCP konekcija od strane čvorova. Iako ovo smanjuje kompleksnost procesa, ovaj scenarij nije poželjan iz pogleda aplikacijske dostupnosti. VM migracija je spojena sa selekcijom topologija drugog i/ili trećeg sloja. Drugi obzir je količina podataka koja se treba prenijeti kad se VM migrira preko mreže, Potencijalno, može biti reda veličine gigabajta, ovisno o VM i uključenog operacijskog okruženja. Dinamična migracija implementira ovaj prijenos u inkrementalnom obliku tako da je zahtjev na mreži raširen. S druge strane, hladna migracija pri kojoj se suspendira VM i migrira, može izazvati nagli udar podataka u mreži dovodeći do problema performansi aplikacije za druge virtualne mašine i fizičke komponente. Oblikovanje količine podataka koja može biti poslana u određenom periodu vremena, rezervacija pojasa i pravilna orkestracija su jako poželjne u ovim situacijama [22].

5.6.1. Mrežni izazovi u IaaS sloju

Postojeći mrežni protokoli i arhitekturi poput *Spanning Tree Protocol* (STP) mogu ograničiti skalu, latenciju, propusnost i VM migraciju u *cloud* mrežama. S toga su predloženi otvoreni standardi i vlasnički protokoli kako bi se adresirali mrežni problemi *cloud computinga*. Dok postojeće stablaste mreže na trećem sloju omogućavaju pristup adresnim zahtjevima za visoko virtualan *cloud* podatkovni centar, postoji više industrijskih standarda koji poboljšavaju oznake drugog sloja mreže, koristeći *Transparent Interconnection of Lots of Links* (TRILL), *Shortest Path Bridging* (SPB) ili sustave bazirane na SDN (eng. *software-defined networking*) konceptu [22].

TRILL je prosljeđujući protokol drugog sloja koji operira unutar jedne IEEE 802.1 *Ethernet broadcast* domene. On zamjenjuje STP protokol koristeći IS-IS (eng. *Intermediate System to Intermediate System*) usmjeravanje za distribuiranje informacije o stanju linka i kalkuliranje najkraćeg puta kroz mrežu. Koristi se IS-IS jer je protokol drugog sloja i ne zahtjeva IP za prijenos okvira. TRILL podatkovni paketi i IS-IS rutirajući paketi se razmjenjuju između usmjernih premosnika (eng. *Routing Bridges – RBridges*) [27]. Drugim riječima,

[9] TCP je konekcijski protokol transportnog sloja OSI RM čija je uloga posluživanje paketa u danom trenutku, kontrola toka i pouzdana isporuka paketa [54].

informacije o povezanosti se prenose kroz cijelu mrežu tako da svaki usmjerni premosnik zna sve o drugima i njihovim vezama. Postoji mnogo prednosti TRILL-a, poput unaprijeđene skalabilnosti i smanjenja problema vezanih uz MAC adresne tablice [28]. Glavni cilj TRILL-a je enkapsulacija svakog *Ethernet* okvira unutar druge omotnice (eng. *envelope*) koja se ponaša kao omotnica trećeg sloja i onda je enkapsulirani okvir usmjeren. Ovo omogućava da se kreira relativno velik *cloud* na drugom sloju s ravnim adresnim prostorom tako da se čvorovi mogu pomicati unutar *clouda* bez promijene njihove IP adrese [1].

SPB je protokol drugog sloja (standard IEEE 802.1aq) koji adresira iste osnovne probleme kao i TRILL, ali s drugačijim pristupom. SPB je protokol koji koristi stablo najkraćeg puta – SPT (eng. *shortest path trees*) kao alternativu rasprostranjenog stabla STP-a. SPT garantira da će promet uvijek biti proslijeđen preko najkraćeg puta između dva premosnika. IS-IS link state protokol je korišten unutar SPB regije za razmjenu informacija između premosnika kako bi izračunali SPT. SPT se računaju za svaki izvorni premosnik u SPT regiji prema svim drugim premosnicima [27].

Unutar SPB protokola se nalaze dva modela za višestaznog premošćivanja: *shorthest path bridging* VLAN (SPBV) i SPB *Mac-In-Mac* (SPBM). Obadvije varijante koriste IS-IS kao link state protokol i obje računaju najkraći put između čvorova. SPBV koristi najkraći put VLAN ID za određivanje dohvatljivosti čvora. SPBM koristi kombinaciju MAC i VLAN ID da odredi dohvatljivosti čvora. Obje varijante pružaju interoperabilnost s STP-om. Za aplikacije u podatkovnim centrima, SPBM je željenija tehnologija [28].

Ključna motivacija iza TRILL-a, SPB-a i SDN-baziranih sustava je relativno ravna priroda topologije podatkovnih centara i zahtjev za prosljeđivanjem paketa preko najkraćeg puta između krajnjih točaka (servera) zbog smanjenja latencije. Drugi sloj (prospajanje) i treći sloj (usmjeravanje) su dvije moguće opcije za mrežu *cloud* infrastrukture. Podatkovni sloj je jednostavnija opcija gdje se za prosljeđivanje koristi *Ethernet* MAC adresa i VLAN informacije. Nedostatak ovog sloja je skalabilnost. Mreža podatkovnog sloja ravna mrežnu topologiju što nije idealno kad postoji velik broj čvorova. Mrežni sloj i subnetiranje omogućavaju segmentaciju za odgovarajuće funkcije.

Mrežna topologija podatkovnog centara je najčešće podešena da odgovara unaprijed definiranim zahtjevima prometa. Na primjer, mrežna topologija optimizirana za promet između servera nije ista kao topologija za promet od i za Interneta. Dizajn topologije također ovisi o tome kako drugi ili treći sloj efektivno iskorištava mrežni kapacitet [22].

Mrežni uređaji i serveri su obično povezani sa statički konfiguriranom fizičkom mrežom što implicitno stvara lokacijsku ovisnost. Na primjer, IP adresa servera je određena temeljem VLAN-a ili subneta kojem pripada. VLAN-ovi i subnetovi se baziraju na konfiguraciji fizičkog porta prospojnika. Dakle, virtualna mašina se ne može jednostavno migrirati kroz mrežu.

Tipična troslojna mreža podatkovnog centra uključuje TOR sloj koji povezuje servere, agregacijski sloj i jezgreni sloj što omogućava povezanost s Internetom. Ova višeslojna arhitektura postavlja osjetnu kompleksnost pri određivanju granica domene drugog sloja, mrežnog prosljeđivanja trećeg sloja i za sloj specifične mrežne opreme [22].

5.6.2. 10 Gb *Ethernet*

10 Gigabit *Ethernet* je grupa tehnologija računalnih mreža koja prenosi *Ethernet* okvire brzinom 10 Gb/s. Za razliku od prošlih *Ethernet* standarda, 10 Gb *Ethernet* definira samo puni dupleks *point-to-point* linkove koji su spojeni mrežni prospojnicima.

Dok se broj priključaka, jezgara i memorijskih utora povećavao po serveru, tipična mrežna povezanost servera se samo povećavala za gigabit u tom vremenu. Mrežna propusnost je postala problem za povećan kapacitet servera. Dodavanjem gigabitne mrežne kartice po potrebi je skupo jer svaka nova mrežna kartica pridodaje potrošnji energije, cijeni kabela i kompleksnosti te povećav broj portova za pristupni sloj prospojnika [29].

Drugi problem poslužitelja koji koriste četiri do osam Gigabitni *Ethernet* je razmnožavanje portova. Nije neobično za servere da budu konfigurirani s do osam mrežnih kartica koristeći portove za svaku funkciju. Konfiguracija sa 6 portova danas više nije dovoljna za radne zadatke. Ciscova testiranja su pokazala da četiri virtualne mašine pokrenute na dvo-priključnom (eng. *socket*) serveru s dvije jezgre može vrlo lako zasititi četiri Gigabitne *Ethernet* linkove [29].

Jedno od rješenja za propusnosti i razmnožavanje portova je prijelaz na 10 Gigabitni *Ethernet*. On pruža dodatne oznake nekim prospojnicima i mrežnim karticama koji podržavaju ujedinjenje više fizičkih kanala u jedan, uključujući sposobnost stvaranja prioritetnih klasa za iSCSI (eng. *internet Small Computer System Interface*), NFS (eng. *Network File System*) i FCoE promet [29].

FCoE je izravna enkapsulacija *Fibre Channel* u *Ethernet* koji očuva karakteristiku bez gubitka i modele upravljanja izvorne *Fibre Channel* mreže. FCoE prenosi *Fibre Channel* direktno iznad *Etherneta* dok je neovisan o *Ethernet* prosljeđivanju. FCoE specifikacija protokola zamjenjuje FC0 i FC1 slojeve *Fibre Channel* stog s *Ethernetom*. S FCoE, Fibre Channel postaje još jedan mreži protkol pokrenut na Ethernetu uz tradicionalni IP promet. FCoE operira iznad Ethernet protokolnog stoga, a iSCSI iznad TCP i IP. Posljedica toga je to da FCoE nije moguće usmjeravati na IP sloju. Računala se mogu povezati na FCoE preko adaptera konvergirane mreže – CNA (eng. *Converted Network Adapters*) koji sadrže Fibre Channel adapter – HBA (eng. Host Bus Adapter) i Ethernet mrežnu karticu [30]. FCOE je podržan u nekim 10 Gb *Ethernet* mrežnim karticama, ali sve podržavaju NFS i iSCSI.

Dakle, umjesto korištenja 4 ili 8 Gb *Ethernet* mrežnih kartica u svakom serveru, organizacije sada mogu razviti samo dvije 10 Gb *Etheret* mrežne kartice i postići potpunu redundanciju za dostupnost. Nadalje, 10 Gb *Ethernet* pruža povećanu skalabilnost koja se očituje u povećanju propusnosti po virtualnoj mašini te podršku mrežne pohrane. Mrežnu pohranu zahtijevaju sofisticirane tehnologije poput VMware Ha i VMotion kako bi se omogućio pristup virtualnim diskovima s bilo kojeg servera [29].

Višestruke virtualne mašine na jednom serveru mogu brzo opteretiti Gigabit Ethernet link, a više linkova može povećati cijenu. S toga, je Cisco doveo novu terminologiju, nazvanu VN-Link. VN-Lik je virtualni link između virtualne mašine i fizičkog sučelja servera. Ova implementacija omogućava operativnu dosljednost svake individualne mašine skupa s njenom

prenosivosti, s toga se mreža i sigurnosne police prenose s virtualnim mašinama po podatkovnom centru [31].

5.7. Virtualizacija infrastrukture *Cloud Computinga*

Virtualizacija obično označava stvaranje virtualne mašine koja može virtualizirati sve fizičke resurse, uključujući procesore, memoriju i mrežnu povezanost. S virtualizacijom, fizički hardver resurse mogu biti podijeljeni na jednu ili više virtualnih mašina. Postoje tri aspekta kako bi se zadovoljila virtualizacija. Prvo, virtualizacija treba omogućiti ekvivalentnu okolinu za pokretanje aplikacije naspram originalnog sustava. Ako aplikacija pokaže drugačije ponašanje pri virtualizaciji, moguće je da okruženje ne ispunjava uvjete. Virtualizacija također treba omogućiti sigurnu kontrolu virtualnih resursa. Održavanje potpune kontrole nad resursima je bitno za zaštitu podataka i resursa na svakom virtualnom okruženju od bilo kakvih opasnosti ili interferencije performansi u podijeljenim fizičkim resursima. Virtualizacija često dovodi do degradacije performansi zbog dodatnih zadataka virtualizacije, ali dobar performans bi trebao biti ostvaren s softverskom i hardverskom podrškom. S ovim zahtjevima, efikasna virtualizacija je zagarantirana. U idućim poglavljima objašnjene su različite vrste hipervizora s implementacijskom razinom virtualizacije [32].

5.7.1. Hipervizor

Kako bi se razumjela virtualizacija, potrebno je prvo pobliže objasniti hipervizor. Hipervizor omogućava komunikaciju između hardvera i virtualne mašine tako da virtualizacija postigne svoju razinu apstrakcije. Hipervizor je originalno nazvan monitor virtualne mašine – VMM (eng. *Virtual Machine Monitor*). Ova dva termina su obično smatrani sinonimima, ali prema nekim izvorima, VMM je softver koji upravlja procesnim jedinicama, memorijama, U/I podatkovnim prijenosima i instrukcijama danim virtualnom okruženju. Hipervizor se može odnositi na operacijski sustav (OS) s VMM [32].

Tipično, hipervizor se može podijeliti na tip 1 i tip 2 hipervizora temeljem različitih razina implementacije. Tip 1 se nalazi na hardveru i komunikacija između hardvera i virtualne mašine je direktna. Domaći operacijski sustav nije potreban u ovom tipu jer je pokrenut direktno na fizičkoj mašini. Zbog ovog razloga, ponekad se naziva „hipervizor čistog metala“ (eng. *bare metal hypervisor*). U ovu kategoriju spadaju: VMware vSphere, Microsoft Windows Server 2012 Hyper-V i *open-source* virtualna mašina na bazi kernela – KVM (eng. *Kernel-based Virtual Machine*) [32].

Tip 2 hipervizor se nalazi na operacijskom sustavu za lakše upravljanje virtualnom mašinom s podrškom hardverske konfiguracije od operacijskog sustava. Dodatni sloj između hardvera i virtualne mašine u tipu 2 stvara smanjenu efikasnost naspram tipa 1 [32]. U slučaju da operacijski sustav doživi kvar, svi krajnji korisnici su pogođeni. Softverska virtualizacija pruža bolu kompatibilnost s hardverom nego u tipu 1, jer je operacijski sustav zaslužan za upravljačke programe hardvera, a ne hipervizor [33]. U ovoj kategoriji su VirtualBox i VMware Workstation. Termini domaća i gostujuća mašina (ili domena) se koriste u hipervizoru kako bi se objasnile različite uloge [32].

Domaća mašina (domena) sadrži hipervizor za upravljanje virtualnim mašinama, a gostujuća mašina (domena) označava položaj svake virtualne mašine na domaćoj mašini u

sigurnom i izoliranom okruženju s vlastitom logičkom domenom. S ovim različitim ulogama, hipervizor pruža resursna ograničenja za više virtualnih mašina na istoj fizičkoj mašini. Drugim riječima, hipervizor je softverski sloj koji stvara virtualno okruženje s virtualnom procesnom jedinicom, memorijom i ulazno-izlaznim jedinicama preko apstrakcije temeljnog hardvera. Virtualna mašina tipično označava enkapsulirani entitet koji uključuje operacijski sustav i aplikacije pokrenute na njemu [32].

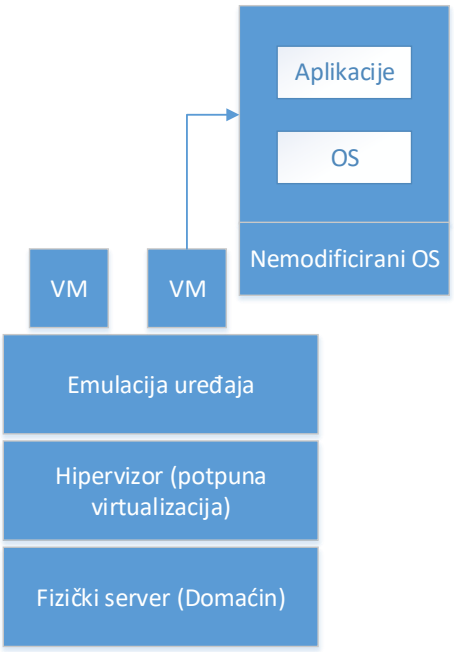
5.7.2. Vrste virtualizacije

Prva vrsta virtualizacije je potpuna virtualizacija. Potpuna virtualizacija je tehnika u kojoj je potpuna instalacija jedne mašine pokrenuta na drugoj. Ova virtualizacija podržava različite operacijske sustave, ali zahtjeva specifičnu hardversku kombinaciju. Hipervizor je u direktnoj interakciji sa serverskim CPU-om (eng. *Central Processing Unit*) i diskom kao što je prikazano na slici 8. U ovoj virtualizaciji svaki virtualni server je potpuno nesvjestan ostalih virtualnih servera koji su trenutno pokrenuti na fizičkoj mašini [33].

Potpuna virtualizacija omogućava virtualizaciju bez modificiranja gostujućeg operacijskog sustava. U x86 arhitekturi, rukovođenje s instrukcijama je ključni faktor virtualnog hardvera. VMware nudi binarni prijevod zahtjeva operacijskog sustava tako da virtualizacija instrukcija može biti završena bez potpore hardvera ili OS. Ovu tehniku također koristi Microsoft Virtual Server [32].

Nadalje, potpuna virtualizacija koristi emulaciju uređaja kao što se vidi na slici 8. Višestruke VM pokrenute na jednoj fizičkoj mašini najčešće komuniciraju koristeći softverski premosnik ili prespojnik pružen od strane hipervizora. U ovom slučaju, funkcionalnost mrežnog uređaja je emuliran virtualizacijskim softverom. Nadalje, fizički server se može pretvoriti u mrežni uređaj koristeći softversku emulaciju. Postoji više softverskih rješenja koji nude takvu funkcionalnost, na primjer *Quagga* i *Open vSwitch* (OVS). Emulacija fizičkog uređaja obično predstavlja dodatni *overhead*[10] [34].

[10] *Overhead* je kombinacija dodatnog ili indirektnog računalnog vremena, memorije, propusnosti ili drugih resursa da se obavi zadatak [56].

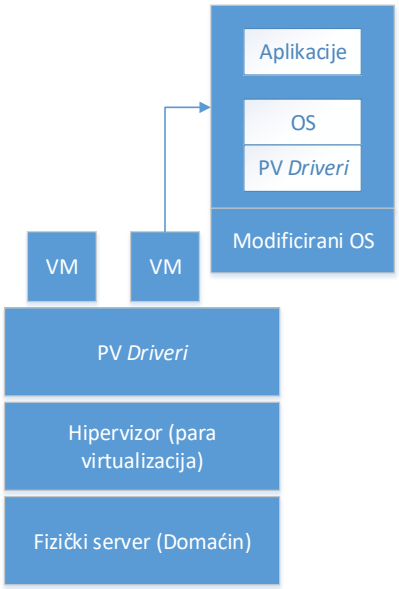


Slika 8. Potpuna virtualizacija

Izvor: [33]

U takozvanoj paravirtualizaciji (PV) koja je prikazana na slici 9, gostujući operacijski sustav je svjestan svoje virtualnosti. Zbog ove dodatne informacije, gostujući operacijski sustav može smanjiti *overhead* komunikacije s fizičkim uređajima [33].

Inicijalno, Xen grupa je razvila paravirtualizaciju za podršku visoke performanse i izolacije snažnih resursa s malim modifikacijama gostujućeg operacijskog sustava. PV zahtjeva modificirani OS kernel. Xen je koristio odvojene *drivere* uređaja i PV operacijske produžetke u Linux kernelu kako bi se izbjegla emulacija uređaja poput mrežne kartice i *drivera* diska. PV skupa s Kernel modifikacijama na gostujućem OS, nudi poboljšan performans CPU, memorije i U/I virtualizacije pri usporedbi s potpunom virtualizacijom. Paravirtualizacija se također naziva virtualizacija potpomognuta operacijskim sustavom, zbog hipervizorovog znanja o gostujućem operacijskom sustavu. PV podržavaju Xen, UML i VMware [32].



Slika 9. Paravirtualizacija

Izvor: [33]

Sklopovski potpomognuta virtualizacija označava virtualne ekstenzije, poznate pod akceleratorima, koje se dodaju x86 procesorima. Riječ je o prilagodbi koja omogućava izmjenu i detekciju osjetljivih instrukcija u sklopovlju. Nedostatak ovakve virtualizacije je zahtjev za posebnim fizičkim strojem koji bi povećao učinkovitost rada u virtualnoj okolini [32].

Nadalje, virtualizacija na razini OS-a ne koristi nikakav hipervizor. Zadaća domaćeg OS je obavljanje svih funkcija potpuno virtualnog hipervizora. Gostujući server mora pokretati isti OS zbog homogenog okruženja [33].

5.7.3. Virtualni privatni server

Virtualni privatni server (eng. *Virtual Private Server*) je virtualna mašina koja se nalazi na nekom računalu. Svaki virtualni server može pokretati OS i svaki se server može restartirati. Korisnici pristupaju sustavu i mogu instalirati većinu softvera koje podržava OS. Za mnoge svrhe jednaki su fizičkim serverima, ali prednost je lakše kreiranje, konfiguracija, povoljnost i podjela hardvera s drugim VPS-ima. Upravo zbog podjele hardvera, njegove performanse mogu patiti, ovisno o opterećenju ostalih VPS-a [33].

Nakon pokretanja fizičkog servera, pokreće se program za pokretanje virtualnih poslužitelja u virtualnoj okolini. Virtualni serveri se dižu iz slike diska (eng. *disk image*) i nemaju pristup hardveru. Kod softverske virtualizacije, virtualne mašine dijele isti kernel i zahtijevaju iste resurse, pa je takva virtualizacija pogodna za *web hosting*. Pri hardverskoj virtualizaciji, nisu moguće nikakve izmjene do ponovnog pokretanja softvera. Takvo okruženje je sigurnije, a princip takvog rada primjenjuje *Microsoft Virtual Server, VMware ESX Server* i *Xen*. Dinamički VPS je onaj koji se može mijenjati tijekom rada, a upravo on se naziva *cloud server*. Njemu se mogu dodavati dodatni hardverski resursi i moguća je migracija na drugi hardver [33].

5.7.4. Virtualno umrežavanje

Tipični fizički domaćin *cloud* podatkovnog centra ima hipervizor koji omogućava pokretanje različitih virtualnih mašina na istom hardveru. U cilju boljeg povezivanja i interoperabilnosti između sustava i mrežnih elemenata, virtualizacija unutar *cloud* infrastrukture postaje čest zahtjev. Tako je hipervizor povezan s virtualnim prospojnikom. Ovaj uređaj prospaja promet drugog sloja između virtualnih mašina pokrenutih na istom fizičkom serveru. Virtualni prospojnik uči o MAC adresama na različit način od tradicionalnih prospojnika jer prethodni smatra po *defaultu* da se svi okviri s nepoznatom odredišnom MAC adresom trebaju proslijediti preko *uplinka* na fizički vanjski prospojnik. Ovo ponašanje može potencijalno stvoriti neke sigurnosne prijetnje poput *packet sniffing*[11]. Nadalje, virtualni prospojnik može prospajati promet između virtualnih mašina prema unaprijed određenim policama koje mogu kontrolirati i promet virtualne lokalne mreže – VLAN (eng. *Virtual Local Area Network*) [1].

Nedostatak virtualnih prospojnika je potencijalno nastajanje ozbiljnih praktičnih problema s tradicionalnim mrežnim arhitekturama. Jedan od njih je konfiguracija VLAN-a: svaki put kad se virtualna mašina migrira na drugi fizički server, potrebno je ponovno konfigurirati VLAN kroz različite prospojnike. Ova koordinacija između prospojnika je kompleksna s velikom latencijom, a nekad je čak i nemoguća ako su prospojnici od različitih prodavača. Potencijalno rješenje za ovaj problem je odvajanje kontrolnih funkcija od mrežnog prosojnika i njihovo postavljanje na pristupne kontrolne servere. Ovo odvajanje podupire softverski definirana mreža – SDN (eng. *Software Defined Networking*) i *OpenFlow* protokol. Neki od nedostataka ovih novih prijedloga su: LAN *overhead*, nedostatak robusnosti kontrolnog servera i VLAN konfiguracija u agregacijskim i jezgrenim prospojnicima [1].

Virtualni LAN (VLAN) je grupa hostova s zajednički interesom da se logički spoje zajedno ispod jedne *broadcast* domene neovisno o njihovoj fizičkoj povezanosti. S obzirom da su VLAN-ovi logički entiteti, odnosno da su konfigurirani u softveru, oni su fleksibilni u smislu mrežne administracije, upravljanja i konfiguracije. Nadalje, VLAN pruža povećanu razinu sigurnosti, izolacije i cjenovno su povoljni. Tipičan VLAN je konstrukt drugog sloja, iako postoje implementacije i na dugim slojevima. Svi okviri u VLAN-u nose zajednički VLAN ID u svojim MAC zaglavljima i svi VLAN prospojnici koriste odredišnu MAC adresu i VLAN ID za prosljeđivanje okvira. Ovaj proces je poznat kao „bojanje okvira“ [34].

[11] *Packet sniffing* je presluškivanje prometa na lokalnoj mreži što omogućava hakerima pristup osobnim podatcima korisnika [55].

6. Analiza mrežne infrastrukture koncepta *Cloud Computing*

U ovom poglavlju obavljena je analiza mrežne infrastrukture *Cloud Computinga*. Ponajprije, objašnjene su razlike u potrebi mrežne infrastrukture za različite razvojne modele i modele dostave *Cloud Computinga*. Nakon toga, prikazan je način na koji javni pružatelji *Cloud Computing* usluge iskorištavaju svoju infrastrukturu, koje usluge nude te koja je razlika između pojedinih. Analizirana su tri pružatelja: Amazon, Microsoft i Google.

Unatoč prednostima C*loud Computinga,* prelazak na njega može biti težak. Organizacije trebaju biti svjesne mnogih problema i jedinstvenih zahtjeva svakog razvojnog modela.

Trendovi su pokazali drastičan porast prijenosa multimedijskog sadržaja preko Interneta zadnjih deset godina. Tako je video postao ključna poslovna aplikacija za komunikaciju korisnika, partnera i zaposlenika, a jako je osjetljiva na latenciju. S toga je ključno da mreža može prepoznati i skalirati video dok se prenosi preko *clouda*. U privatnim i hibridnim *cloud* okruženjima, mreža treba biti prilagođena dostavljanju optimalnog iskustva [35].

Nadalje, WAN performanse su novi problem jer aplikacijske performanse omogućene od strane mreže direktno utječu na korisnike. Izazov je dati konzistentan, visoko kvalitetan performans aplikacijama kad se pružaju usluge preko *clouda*. Kako sve više upravljačke lokacije postaju korisnici, a ne pružatelji, WAN optimizacija je nužna. U privatnim i hibridnim *cloudovima*, mreža treba omogućavati optimizaciju WAN i *cloud* resursa [35].

Prema [35], većina IaaS arhitekata kreiraju mrežnu infrastrukturu na pretpostavkama da korisnici žele *broadcast* domenu drugog sloja i da je za postizanje visoke iskoristivosti potrebna neograničena mobilnost VM preko cijele infrastrukture.

Ove dvije pretpostavke ograničavaju opcije na rastegnute VLAN-ove preko cijelog podatkovnog centra. Pri implementaciji s tradicionalnim prospojnicima podatkovnog centra, broj VLAN-ova koje podržavaju prospojnici (1000 do 4000 ovisno o modelu), direktno određuju broj korisnika koji se mogu podržati. Amazon Web Services (AWS) je pokazao da korisnicima ne treba nužno domena drugog sloja te da nije potrebna neograničena mobilnost [35].

6.1. Amazon EC2

Amazon Web Services (AWS) je set *cloud* usluga, nudeći računalstvo u oblaku, pohranu i druge funkcionalnosti koje omogućavaju organizacijama i pojedincima razvoj aplikacija i usluga. AWS usluge su dostupne preko HTTP-a , koristeći REST i SOAP protokole [13].

Nije novost da internetski i softverski divovi dizajniraju svoj hardver kako bi povećali efikasnost i stekli prednost nad konkurencijom. Ipak, AWS je u svojim dizajnima otišao najdalje. AWS ne samo da je dizajnirao vlastite usmjeritelje, čipove, servere pohrane i računalne servere, nego je dizajnirao i vlastitu brzu mrežu [36].

Usprkos tome što su industrijski standardi 10 GbE (engl. *Gigabit Ethernet*) i 40 GbE, AWS je standardizirao na 25 Gb *Ethernet*. Razlog tome, prema AWS-u, je taj što je 25 Gb približan cijeni 10 Gb, a ostvaruje dovoljne brzine da bude isplativiji od 40 GbE. Nadalje, AWS

usmjeritelji pokreću Broadcom Tomahawk ASIC[12] sa sedam milijardi tranzistora i 128 *portova* 25 GbE za ukupnu protočnost od 3.2 terabita. Još jedan ključni dio AWS mrežne strategije je softverski definirana mreža gdje mrežni administratori mogu promijeniti i upravljati mrežom kroz sučelja [36].

AWS koristi opremu s prilagođenim ugrađenim softverom za podatkovne centre što znači da u slučaju kvara izvan centra, postrojenje nastavlja raditi, a ako je kvar unutra, napor nije izgubljen. AWS-ovi posljednji prilagođeni serveri pohrane spremaju 11 petabajta podataka na 1100 diskova postavljenih u jedno postolje standardne veličine 42U. AWS podatkovni centri su povezani 100 Gb privatnom mrežom koju kontrolira isključivo AWS. AWS prilagođeni računalni serveri, zauzimajući jedan utor u postolju, nisu gusto postavljeni što povećava terminalnu efikasnost [36].

Amazon Elastic Comupte Cloud (Amazon EC2) koristeći IaaS platformu, omogućava korisnicima korištenje virtualnih resursa na *cloud computing* platformi te stvaranje aplikacija na njoj. Korisnici imaju vlastite virtualne računalne mašine koje se naplaćuju prema korištenju i servisnom planu. Amazon koristi Elastic Compute Unit (ECU)[13] kao apstrakciju računalnih resursa. Svaka EC2 računalna jedinica se računa kao ekvivalent mašini s 1-1.2 GHz 2007 Opteron ili 2007 Xeon procesorom [37].

EC2 pruža *cloud* korisnicima pokretanje i upravljanje serverskim instancama u podatkovnim centrima koristeći grafičko sučelje ili dostupne alate. EC2 instance su virtualne mašine pokrenute na Xen virtualnom uređaju. Nakon stvaranja i pokretanja instance, korisnici mogu učitavati softver i činiti promjene na njemu. Kad se promjene naprave, mogu biti spojene u novu sliku mašine. Identična kopija može biti pokrenuta u bilo koje vrijeme. Korisnici imaju pa skoro potpunu kontrolu nad cijelom grupom softvera na EC2 instancama koje za njih izgledaju kao hardver. S druge strane, ova oznaka čini osjetno težim za Amazon da ponudi automatsko skaliranje resursa [13].

EC2 pruža mogućnost postavljanja instanci na različite lokacije. EC2 lokacije su sastavljene od regija i dostupnih zona. Regije su izolirane jedne od drugih s naumom minimiziranje mrežne udaljenosti između korisnika i usluga. U svijetu, postoji 16 geografskih regija, a još dvije bi trebale postati *online* u 2017. Komunikacija između različitih regija je preko javnog Interneta tako da korisnici trebaju odabrati prikladnu enkripciju za zaštitu svojih podataka [37]. Regije se sastoje od jedne ili više dostupnih zona. Dostupne zone su različite lokacije koje su stvorene kako bi bile izolirane od kvarova u drugim dostupnim zonama te da bi pružile jeftinu mrežu s niskom latencijom s drugim dostupnim zonama u istoj regiji [13].

Ipak, pored mrežnih problema, ova mogućnost je pružila hakerima da koriste EC2 mašine locirane u drugim geografskim zonama za pokretanje malicioznih napada na organizacijske infrastrukture. Nadalje, problem se može javiti i pri regulatornim akcijama korisničkih podataka spremljenih na *cloud*. Na primjer, državne organizacije u Sjedinjenim Američkim Državama

[12] ASIC (engl. *Application-specific integrated circuit*) je mikročip dizajniran za specijalnu aplikaciju poput pojedinačnog tipa prijenosnog protokola [57].

[13] ECU je Amazonova metrika za korištenje CPU kapaciteta. ECU omogućava relativnu mjeru procesne snage Amazon EC2 instance [35].

ne mogu pohraniti podatke u podatkovne centre locirane u Europi. Europske države imaju stroga pravila o zaštiti podataka, posebno o EU stanovnicima i njihovim privatnim podatcima [37].

EC2 slike mašina - AMI (eng. *Amazon Machine Image*) pruža informaciju potrebnu za pokretanje instance. AMI se specificira kad se pokrene instanca i može se pokrenuti koliko je god potrebno instanci. Također je moguće pokrenuti instance s različitih AMI [38]. Slike mašina se spremaju i ponovno pokreću s Amazon Simple Storage Service (Amazon S3). S3 pohranjuje podatke kao objekte koji su pohranjeni u „kante" (eng. *buckets*). Svaki objekt se sastoji od 1 bajta do 4 gigabajta podataka. Objektna imena su nužno URI imena puta. Kante moraju biti eksplicitno kreirane prije nego što se mogu koristiti. Kanta može biti pohranjena u jednoj ili više regija, a korisnici biraju regiju kako bi optimizirali latenciju, troškove i slično [13].

Amazon Virtual Private Cloud (VPC) je siguran i besprijekoran most između postojeće IT infrastrukture tvrtke i AWS *clouda*. Amazon VPC pruža tvrtkama povezivanje njihove postojeće infrastrukture sa setom izoliranih AWS računalnih resursa preko virtualne privatne mreže [13].

Za cloud korisnike, Amazon CloudWatch je koristan alat za upravljanje koji koristi sirove podatke iz partnerskih AWS usluga poput Amazon EC2 i onda procesuira informaciju u čitljivu metriku. Metrike o EC2 uključuju npr. iskorištenje centralne procesne jedinice, mrežnih U/I bajtova, učitavanje diska i slično [13].

Amazonova elastična IP adresa je svojstvo slično statičnoj IP adresi u tradicionalnim podatkovnim centrima, ali s jednom ključnom razlikom. Korisnik može programski mapirati elastičnu IP adresu bilo kojoj instanci virtualne mašine bez pomoći mrežnog administratora ili bez čekanja DNS-a. U ovom smislu, elastična IP adresa pripada računu, a ne instanci virtualne mašine. Ona postoji sve dok se eksplicitno ne ukine te je povezana s računom čak i kad nije povezana s instancom [37].

Amazon je prvi prodavač široke raznolikosti javnih *cloud* usluga koje su prihvaćene od strane šire publike, pa stoga mnogi istraživači koriste *Amazon Clouds* za testiranje i evaluaciju performansi *cloud* usluga. Već pri ranim istraživanjima, autori su otkrili da *Amazon* usluge nude praktičnu alternativu za organizacije koje žele iskoristiti *Cloud comuting*, a s druge strane su bili zabrinuti za dosljednost performansi i sigurnosne probleme [39].

6.1.1. Analiza performansi Amazon EC2

Testiranjem podjele centralne procesne jedinice koristeći softver, rezultati su pokazali da male instance uvijek dijele procesor s drugim instancama. U većini slučajeva, male instance dobiju 40% - 50% fizičkog CPU-a. Smatra se da Amazon EC2 koristi isključivo policu zakazivanja virtualne mašine kako bi se kontrolirao računalni kapacitet za instance. Čak i kad nema drugih virtualnih mašina pokrenutih na istom serveru, male instance i dalje ne mogu koristiti više od 50% procesora. Ipak, srednje instance dobiju 100% dijeljenja CPU-a i postoji samo 20% slučajeva gdje su dobile 95% što može biti izazvano zbog komunikacije između korištenog softvera za mjerenje i *kernel* procesa [39].

Nadalje, pri mjerenju TCP/UDP propusnosti korišteno 750 pari malih instanci i 150 pari srednjih instanci na različitim mrežnim lokacijama. U vremenskom testiranju, mjeren je TCP/UDP propusnost 6 pari malih instanci i tri para srednjih instanci kontinuirano tijekom 150 sati. Istraživanje je pokazalo da mreža Amazon EC2 podatkovnog centra nije teško opterećena i EC2 instance mogu ostvariti TCP propusnost višu od 500 Mb/s u većini slučajeva. Medijan TCP/UDP propusnosti srednjih instanci je 760 Mb/s. Ipak, TCP propusnost malih instanci je mnogo niža nego UDP propusnost (570 Mb/s naspram 770 Mb/s). Mjerena nestabilna propusnost TCP/UDP nastaje virtualizacijom krajnjeg *hosta* [40].

6.1.2. Analiza Amazon EC2 u svrsi računalstva visokih performansi

Računalstvo visokih performansi – HPC (eng. *High Performance Computing*) za znanstvene aplikacije zahtjeva velike količine računalnih resursa i pohrane skupa s mrežnom propusnošću. *Cloud* ideja omogućava efikasan način eliminacije posjedovanja klastera. Prema istraživanju i kvantitativnoj analizi, Amazon EC2 podržava razumni performans za male na-zahtjev HPC aplikacije. Ipak, evaluacija u drugi radovima je pokazala da je EC2 sporiji nego tipični Linux klaster srednjeg ranga i mnogo sporiji od modernog HPC sustava [39].

Nadalje, istraživanja su pokazala da su loše performanse mreže nastale zbog virtualizacije U/I overheada i dijeljenje procesora glavno ograničenje performansi i skalabilnosti u Amazon EC2. Kako bi se riješili ovi problemi, Amazon je dodao Cluster Compute (CC) platformu u EC2 cloud usluge. CC platforma je skupina tipova instanci koje zapošljavaju snažnije centralne procesne jedinice i brze mreže za podupiranje HPC aplikacija. Analizom dvaju CC instanci, otkriveno je da se skalabilnost HPC aplikacija oslanja na performans komunikacija što ovisi o mreži i njenom efikasnom podržavanju virtualnog sloja [39].

Rezultati usporedbe performansi Amazon EC2 i Microsoft Azurea pri HPC je pokazalo slabije performanse obadva *clouda* naspram stvarnih računala, jer je performans ograničen brzinom konekcija. Azure je generalno pokazao bolji performans naspram EC2 zbog boljih konekcija. Aplikacije koje se većinom fokusiraju na čisto procesiranje su ipak bile brže na EC2. S druge strane, Azure je imao veću efikasnost. Zaključak je, da za HPC ne postoji idealno *cloud* rješenje, nego da se odabir treba temeljiti naspram potrebama pojedine aplikacije [41].

6.2. Microsoft Azure

Windows Azure je Microsoftova aplikacijska platforma za javni *cloud*. To je PaaS platforma koja radi kao operacijski sustav i pruža usluge za razvoj aplikacija i pohranu podataka.

Microsoftova arhitektura *cloud* servera je temeljena na modularnoj visoko-zgusnutoj šasiji koja omogućava efikasnu podjelu resursa preko višestrukih čvorova. Jedna 12U šasija može spremiti do 24 plošna servera (računalna ili pohrandbena), gdje su dva plošna servera spremljena u svaki 1U utor. Svaki računalni server ima 10 Intel Xeon E5-2400 procesora. Microsoft Azure koristi 10 GbE ili 40 GbE [42].

Windows Azure platforma se sastoji od tri komponente i svaka od njih pruža specifičan set usluga *cloud* korisnicima. Windows Azure pruža Windows-bazirano okruženje za pokretanje aplikacija i pohranu podataka na servere u podatkovnim centrima; SQL Azure pruža

podatkovne usluge na SQL serveru, .NET usluge pružaju usluge distribuirane infrastrukture za *cloud* bazirane i lokalne aplikacije. Windows Azure platformu mogu koristiti aplikacije pokrenute u *cloudu* i aplikacije pokrenute na lokalnim sustavima.

Windows Azure također podržava aplikacije izgrađene na .NET radnom okviru i drugim jezicima podržanim u Windows sustavima (npr. C#). Radni okvir Windows Azurea je mnogo fleksibilniji od Google AppEngine, ali ograničava korisnikov izbor modela pohrane i strukture aplikacije. Stoga, Azure je sredina između aplikacijskih radnih okvira poput AppEnginea i hardverskih virtualnih mašina poput EC2 [3].

Svi fizički resursi, virtualne mašine i aplikacije u podatkovnom centru se nadgledaju softverom. Sa svakom aplikacijom, korisnici mogu učitati konfiguracijsku datoteku koja pruža opis što aplikacija treba. Na osnovu toga, softver odlučuje gdje se nove aplikacije trebaju pokrenuti odabirom fizičkih servera za optimizaciju hardvera [43].

Kao i kod EC2, korisnik odabire koliko instanci želi pokrenuti. Ipak, jednom kad se razvije softver, slika prilagođene instance se automatski pokreće s pokretanjem softvera. Instance su podijeljene u više ažuriranih domena, tako da se pri kritičnim sigurnosnim ažuriranjima ili većim sustavnim ažuriranjima, pojedinačna domena isključi, ažurira i ponovno pokrene. Ovo garantira posjedovanje najnovijeg softvera bez isključivanja kompletnog klastera te može smanjiti vrijeme rada u usporedbi s tradicionalnim EC2 instancama. Azure zadržava fleksibilnost koju ima EC2 dopuštajući prilagodbu instanci ako korisnik ne želi raditi s .NET [44].

6.2.1. Azure usluge pohrane

U ovom poglavlju analiziraju se performansi tri Azure usluge pohrane: grumen (eng. *blob*), tablica (eng. *table*) i red (eng. *queue*). Za svaku uslugu je mjeren maksimalna propusnost u operacijama po sekundi ili MB po sekundi i skalabilnost usluge kao funkcija konkurentnih klijenta. Generalno, skalabilnost je problem za aplikacije na velikoj skali s više konkurentnih klijenata ako podatci nisu prikladno podijeljeni ili replicirani [45].

Grumen usluga u Windows Azuru omogućava hijerarhiju pohrane: računi pohrane imaju više kontejnera koji sadrže jedan ili više grumena. Svaki individualni grumen može pohrani jedan terabajt podataka i povezne metapodatke. Pri analizi jednog grumena od 1 gigabajta, maksimalna propusnost servera za *download* je 393.4 MB/s koristeći 128 klijenata. Nadalje, za *upload* operaciju, maksimalna propusnost je 124.25 MB/s što je promatrano s 192 konkurentna klijenta [45].

Tablica ili *table,* je Azure set entiteta sa svojstvima, gdje svako svojstvo može imati različite tipove i tablica nema definiranu shemu. U ovom smislu, različita je od tablice u relacijskoj bazi podataka. Jedna tablica ima sposobnost pohraniti milijarde entiteta noseći terabajte podataka. Analiza tablice je mjerena s različitim veličinama entiteta: 1KB, 4KB, 16 KB i 64 KB. Za operacije *Insert* i *Query*, performans klijenata je padao kako se povećavala razina konkurencije. *Update* i *Delete* testovi su pokazali još drastičniji pad performansa s povećanje broja klijenata. Ove dvije operacije su imale visoku inicijalnu propusnost s jednim klijentom, ali s osjetnim usporenjem kad se broj klijenata povećao [45].

Red u Windows Azure služi kao komunikacijski objekt između mrežnih uloga (eng. *Web role*) i radnih uloga (eng. *Worker role*). Mrežna uloga je konfigurirana za pokretanje web aplikacija razvijenih u programskim jezicima koje podržava *Internet Information Services* (IIS) poput PHP-a i ASP.NET, a radna uloga je bio koja uloga u Azureu koja pokreće aplikacije usluge koje generalno ne zahtijevaju IIS tj. nije instaliran po zadanom [46].

U ispitivanju reda, korišten je red koji se dijeli između 1 do 192 radnih uloga te je promatrana skalabilnost tri operacije: *Add, Peak* i *Receive*. Za svaku operaciju su korištene različite veličine poruke: 512 bajta, 1 KB, 4 KB i 8 KB. Generalno, rezultati su pokazali da operacije *Add* i *Receive* imaju slično ponašanje, a *Peak* poruka je najbrža. Nadalje, za *Add* i *Receive,* maksimalni performans je na 64 konkurentna klijenta. Za *Peak,* ipak, propusnost je bila najbolja sa 192 konkurentna klijenta [45].

6.2.2. Azure računalne usluge

U ovom poglavlju se obrađuje akvizicija instanci i TCP komunikacija između različitih instanci virtualnih mašina.

Vrijeme akvizicije računalnih instanci je kritična metrika za određivanje efikasnost dinamične skalabilnost *cloud* aplikacije. U istraživanju [45], napravljen je test program koji koristi Windows Azure menadžment API za prikupljanje podataka o Azure instancama virtualnih mašina. Za svako pokretanje programa, on je nasumice odabra veličinu virtualne mašine [45].

Nakon prikupljenih podataka u istraživanju [45], od 431 uspješnih pokretanja, postotak neuspjeha pokretanja VM, uključujući testne slučajeve, je 2.6%. Mrežna uloga VM instanci treba dulje vrijeme pokretanja nego radna uloga instanci. Za sve veličine VM, mrežnoj ulozi treba 60 sekundi više nego radnoj ulozi. Takvo mjerenje je očekivano, s obzirom da mrežna uloga zahtjeva kompleksniju inicijalizaciju nego radna uloga (npr. IIS podrška) [45].

Prosječno vrijeme pokretanja male instance radne uloge je oko 9 minuta, dok je prosječno vrijeme mrežne uloge 10 minuta. Najbrže promatrano vrijeme je 7.5 minuta za radnu ulogu i 9 minuta za mrežu ulogu. Za 85% testnih pokretanja, prva mala instanca radne uloge bude spremna unutar 9 minuta i za 80% prva mala instanca mrežne uloge bude spremna unutar 10 minuta [45].

Azure ne poslužuje zahtjeve od više VM u isto vrijeme. Odnosno, postoji usporavanje između vremena kad prva instanca postane dostupna i sljedeće instance. Za obje uloge, promatrano vrijeme između prve i četvrte instance je bilo 4 minute.

Nadalje, performans razvoja aplikacije je funkcija ovisna o veličini aplikacije. Tako je aplikacija od 1,2 MB bila pokrenuta 30 sekundi brže od 5 MB aplikacije. Ipak, pri testiranju pohrana je bila u Azure usluzi pohrane. Ako su aplikacije pohranjene lokalno, vrijeme razvijanja bi bilo mnogo dulje zbog ograničenja propusnosti lokalne mreže [45].

Drugi dio analize računalstva *Windows Azurea* je TCP komunikacija. Naime, *Windows Azure* dozvoljava programerima definiranje TCP i HTTP[14] unutarnjih krajnjih točki za instance virtualnih mašina. Ovakva vrsta komunikacije je čvrsto spojena, radi samo u obliku *point-to-point* i aplikacija treba definirati protokol. Ipak, ona je dobar kompliment, jer unutarnji *portovi* dozvoljavaju virtualnim mašinama da direktno komuniciraju jedna s drugom koristeći TCP/IP *portove* s niskom latencijom i širokim pojasom. S toga je mjereno ovo obilježje Azure virtualnih mašina temeljem tri metrike: latencija, širina pojasa i varijabilnost širine pojasa preko vremena [45].

Prvi eksperiment se sastojao od 20 malih virtualnih mašina. Pola ih je mjerilo latenciju, a pola širinu pojasa. Svaka virtualna mašina je bila sparena s drugom; svaki par je sadržavao server i klijenta. Kako bi se izmjerila latencija, klijent mjeri vrijeme prijenosa informacije od 1 bajta preko TCP kanala. Za mjerenje širine pojasa, klijent je slao 2 GB informacije serveru [45].

Rezultati su pokazali da otprilike 50% vremena latencije iznosi 1 ms, a 75% latencije iznosi 2 ms. Općenito, najčešće će latencija podatkovnog centra sličiti lokalnoj mreži. S druge strane, mjerenja širine pojasa su pokazala da je 50% vremena širina pojasa 90 MB/s ili veća. Pretpostavlja se da je fizički hardver Gigabitni *Ethernet*, koji ima ograničenje 125 MB/s. Dakle, eksperimenti su pokazali da virtualne mašine imaju dobru širinu pojasa, mada je donjih 15% imalo pad performansi na 30 MB/s ili gore [45].

Drugi eksperimentalni sklop se sastojao od dvije virtualne mašine, jednog klijenta i jednog servera. Ove dvije virtualne mašine su prenosile ista 2 GB podataka, ali na regularnih intervalima od pola sata po nekoliko dana. Rezultati su pokazali da većinom širina pojasa prelazi 80 MB/s, ali postoje i trenutci kad je spor popit 10 MB/s. Zbog toga što su korištene male instance, resursni menadžer će najvjerojatnije premjestiti druge instance na isti fizički hardver što može osjetno utjecati na promatranu širinu pojasa. Iako je sličnost s lokalnom mrežom još uvijek velika, postoji velika varijabilnost. TCP je obilježje čija je razina varijabilnosti najveća za Windows Azure platformu [45].

6.3. Google App Engine

Google App Engine je PaaS platforma za razvoj i održavanje aplikacija preko iste infrastrukture koja omogućava Google web aplikacije. Ova platforma podržava programske jezike poput *Pythona* ili *Jave* [47].

Google App Engine koristi prilagođene usmjeritelje i prospojnike s kapacitetom 128 portova za 10 GbE. Pokrenuta su minimalno dva usmjeritelja po podatkovnom centru što znači da se mreža skalira u terabit po sekundi rasponu. Ovi prilagođeni prospojnici i usmjeritelji su povezani s uređajima za valno multipleksiranje za spajanje s podatkovnim centrima [42].

Iz podatkovnog centra, mreža započinje na postolju gdje su postolja prilagođeni i sadrže 40 do 80 servera. Svako postolje sadrži prospojnik. Serveri su povezani preko 1 Gbit/s Ethernet linka na TOR prospojnik. TOR pospojnici su povezani na gigabitni klaster prospojnik [42].

[14] HTTP (eng. *Hypertext Transfer Protocol*) je ASCII protokol sloja aplikacije OSI RM gdje se interakcija sastoji od jednog ASCII zahtjeva kojeg slijedi odgovor [54].

Sve korisničke aplikacije su pokrenute u sigurnom okruženju – *The Sandbox*. Ovo okruženje ima ograničen pristup osnovnom operacijskom sustavu. Zbog ovih ograničenja, App Engine je u mogućnosti distribuirati aplikacijske web zahtjeve preko više servera što dopušta pokretanje i zaustavljanje servera kako bi zadovoljili zahtjeve prometa. Sandbox odvaja aplikaciju u njeno vlastito zaštićeno okruženje koje je neovisno o operacijskom sustavu, hardveru ili fizičkoj lokaciji web servera [48].

Aplikacija može pristupiti drugim računalima na Internetu kroz pruženi URL, email usluge i API, a druga računala se mogu povezati s aplikacijom stvaranjem HTTP zahtjeva na standardnim portovima. Nadalje, aplikacija ne može upisivati na *file system* i može učitavati samo podatke koji su učitani s aplikacijskim kodom. Aplikacijski kod je samo pokrenut kao odgovor na web zahtjev i mora vratiti odgovor unutar 30 sekundi [48].

NoSQL označava ne-relacijske baze podataka. Ovaj tip baza podataka ne podržava potpuni relacijski model i nedostaje dosljednosti, ali se fokusiraju na performans i skalabilnost. Naime, performans relacijskih baza podataka opada s povećanjem podataka, pa kad veličina podataka dostigne gigabajte i terabajte, teško je upravljati tolikom količinom podataka; mnoge web stranice koje koriste takvu količinu podataka su se suočile s tim problem (npr. YouTube) [49].

Google File Sytem (GFS) je skalabilan, distribuiran podatkovni sustav i kao svi tradicionalni podatkovni sustavi nudi pouzdanost, dostupnost, skalabilnost i performans. Kreiran je od strane Googleovih inženjera kako bi susreo zahtjeve pohrane njihovih aplikacija. GFS se sastoji od više čvorova koji su podijeljeni u dva tipa: glavni čvor i veliki broj *chunk* servera[15]. Svaki dokument je podijeljen u komade određene veličine koji su pohranjeni u *chunk* servere. Svakom komadu glavni čvor dodjeljuje jedinstvenu 64-bitnu oznaku u trenutku kreiranja. Svaki komad je repliciran više put kroz mrežu. Glavni server obično ne sadrži zapravo komade, nego metapodatke povezane s njima [49].

Veličina komada je 64 MB što je veliko pri usporedbi s drugim sustavima pohrane, ali to ima svoje prednosti. GFS je dizajniran za podržavanje velikih struja upisivanja i čitanja, tako da velika blok veličina omogućava klijentu da dobije što više podataka iz istog *chunk* servera. Također, takva veličina omogućava manje metapodatke što omogućava GFS pohranu metapodataka u memoriji za visok performans [49].

Prema rezultatima mjerenja u [50], prosječno vrijeme upisivanja je 30 MB/s, a vrijeme učitavanja je mnogo veće naspram vremena upisivanja što je bilo očekivano s obzirom da se ukupni zadatak sastoji od više upisivanja. Klaster je imao stabilnu razinu očitavanja 750 MB/s, dakle koristi efikasno resurse. Što se tiče metapodataka, svaki individualni server ima 50 do 100 MB metapodataka. S toga je lako obnoviti podatke, potrebno je samo par sekundi za učitavanje metapodataka sa diska [50].

BigTable je distribuirani sustav pohrane za upravljanje strukturiranih podataka raširenih preko tisuću sustava. Razvijen je s podrškom skalabilnosti, visoke dostupnosti i pouzdanosti dok još uvijek održava optimalni performans. BigTable je kombinacija modela relacijskih

[15] Chunk server predstavlja naziv za Googleove servere u usluzi GFS.

podataka i skalabilnosti gdje pruža jednostavan model podataka s dinamičnom kontrolom nad formatom i rasporedom podataka [50].

Set procesa pokreće BigTable softver što se naziva BigTable klaster. U BigTable, podatci su pohranjeni u tablice kao relacijske baze podataka ali ove tablice su višedimenzionalne sortirane mape, organizirane kao red, stupac i vremenska oznaka. Podatak referenciran redom, stupcem i vremenskom oznakom se naziva ćelija. Treća dimenzija - vremenska oznaka, omogućava pohranu više različitih verzija podatka [49].

Analiza rezultata eksperimenata [49], pri usporedbi Google App Enginea i Amazon EC2 je pokazala da GAE ima slabo vrijeme odgovora za mali broj zahtjeva, ali s povećanjem broja konkurentnih zahtjeva, GAE vrijeme odgovora se znatno poboljšalo u usporedbi s EC2. Ovo pokazuje efikasno iskorištenje resursa u GAE te bolju mrežnu propusnost.

Amazon EC2 je bolje funkcionirao pri kratkim strujama upisivanja, ali GAE je imao bolji performans kad su konkurentni zahtjevi dosegli više od 200. GAE pokazuje nekonzistentno ponašanje pri kombinaciji nasumičnih operacija upisivanja i učitavanja. Što se tiče, skalabilnosti, GAE je bio jednoliko ponašanje naspram varijabilnog ponašanja EC2 [49].

Velike varijacije u vremenu odgovora i izvršavanja operacije za GAE pokazuje kako su njegovi fizički resursi različiti u pogledu procesne brzine, dok Amazon EC2 ima homogen hardver. Aplikacije razvijene za GAE nisu kompatibilne s drugim platformama i ne mogu se prenijeti na druge platforme u postojećem obliku. Uz to, pri razvijanju prototipne aplikacije, bilo je potrebno više vremena i linija koda za razvoj GAE prototipne aplikacije u usporedbi s Amazonovim EC2. Uzrok tome je ograničena dostupnost razvojnih resursa za GAE, razvojni radni okviri i tehnologije te pomoćni resursi [49].

Dakle, neki nedostatci GAE su: podržavanje samo dva programska jezika, ograničenje od 30 sekundi na vrijeme izvršenja web zahtjeva, aplikacije nemaju pristup OS-u i hardverskom sloju, pohrana podataka ne dozvoljava objekte veće od 10 MB [49].

U usporedbi s Amazon EC2 koji je IaaS, Google App Engine je PaaS. Amazon EC2 podržava razvoj bilo kakvih aplikacija, dok GAE podržava samo web bazirane aplikacije. GAE podržava samo NoSQL sustav pohrane, dok EC2 podržava oboje NoSQL i SQL baziranu podršku. Amazon Relational Database usluga pruža MYSQL baziranu uslugu baze podataka korisnicima. GAE podržava potpuno automatsko skaliranje i pozicioniranje resursa i korisnik nema kontrolu nad resursima dodijeljenim njihovim aplikacijama. S druge strane, Amazon EC2 ima mogućnost konfiguracije. On pruža korisnicima pokretanje broja instanci gdje svaki ima vlastiti operacijski sustav [49].

GAE se preporuča za korisnike kojima su potrebne ne-transakcijske aplikacije i aplikacije koje ne obavljaju teške analize nad velikim količinama podataka, jer GAE nema podršku za globalne transakcije te je njegov procesni performans lošiji od Amazon EC2. Aplikacije koje zahtijevaju veliku količinu računalstva nisu prikladne za GAE s obzirom da GAE ne dozvoljava ručno dodjeljivanje resursa, dok Amazon EC2 ima ovu osobinu i veliki broj resursa (sa snažnim procesnim mogućnostima) [49].

GAE pruža brzo postavljanje za razvoj i produkciju okruženja u usporedbi s Amazonom EC2. U ovom slučaju, EC2 zahtjeva konfiguraciju resursa poput mreže, instalacija operacijskog sustava i softvera, što može biti kompleksno i teško za održavanje u nekim slučajevima. GAE eliminira problem održavanja operacijskog sustava. Vrijeme razvoja i vrijeme otklanjanja pogrešaka je dulje na GAE nego Amazon EC2. Ovo je bilo očigledno pri razvoju prototipne CRM aplikacije, gdje je bilo potrebno 12.5% (474) više linija koda za razvoj na GAE. Jedan od razloga za povećanje linija koda i vremena je nedostatak kompatibilnosti i podrške za programske knjižnice i radne okvire na GAE. Uz to, postoji ograničena pomoć i podrška dostupna za razvijače [49].

7. Zaključak

Cloud Computing je suvremeni koncept čiji princip iskorištenja i podjele fizičkih resursa mijenja način poslovanja u cijelom svijetu. Njegove usluge su lako dostupne i omogućavaju optimiziranje poslovnih procesa na razini koja prije nije bila izvediva. Ipak, iza svih pruženih usluga, krije se složena mrežna infrastruktura. Tehnologije koje omogućavaju *Cloud Computing* se stalno mijenjaju i nadograđuju kako bi prijelaz s tradicionalnog računalstva na *Cloud Computing* bio sasvim besprijekoran i siguran proces za sve korisnike.

S obzirom da prijelaz na *Cloud Computing* nije jednostavan i postavlja puno izazova pred korisnike, na njemu je da zadovolji mnoge uvjete, odnosno da se što više približi performansama tradicionalnom računalstvu. Pošto je riječ o dijeljenim resursima, jedan od najzanimljivijih uvjeta je skalabilnost. Skalabilnost omogućava prilagodbu *Cloud Computinga* na promjene u opterećenju, a s obzirom da različiti modeli dostave *Cloud Computinga* trebaju različito skaliranje, još uvijek je izazov uspješno i maksimalno skalirati infrastrukturu i usluge.

Razvojem novih protokola osigurana je sigurna i brža komunikacija između podatkovnih centara što je u umreženom svijetu krucijalno. Postoji još mnogo izazova u mrežnoj infrastrukturi *Cloud Computinga*, od kojih su najveći slaba dostupnost potrebnog frekvencijskog pojasa na zahtjev korisnika pri prijenosu podataka.

Uz to, različiti pružatelji usluga javnog *cloud* modela iskorištavaju tu osnovnu infrastrukturu na različite načine, pa i pružaju različite usluge. Prema svim istraživanjima navedenim u radu, moje mišljenje je da Amazon EC2 pruža najbolju mrežnu i računalnu infrastrukturu *Cloud Computinga*. Unatoč svim naprecima u performansama *Cloud Computinga*, stvarna računala još uvijek imaju veću procesnu moć pri računalstvu visokih performansi. Svaki javni *cloud* pružatelj ima svoje prednosti i nedostatke, a korisnik na osnovi svojih zahtjeva uz pomoć znanstveno utemeljenih metoda (primjerice korištenjem metoda višekriterijskog odlučivanja) analiza može donijeti odgovarajuću odluku.

Na temelju provedene analize, koncept *Cloud Computinga* predstavlja vrlo aktualno područje za istraživanja i provedbu usmjerenih analiza s obzirom na različitost primjene. *Cloud Computing* je omogućio iskorištenje resursa koji su prije bili nedostupni i neiskorišteni te će u budućnosti imati veliku ulogu pri ostvarenju poslovanja.

Literatura

[1] Moura J., Hutchison, D.: *Review and Analysis of Networking Challenges in Cloud Computing*, Institut telekomunikacija, Sveučilište u Lisabonu, 2015.

[2] URL: https://www.ibm.com/blogs/cloud-computing/2014/03/a-brief-history-of-cloud-computing-3/. (pristupljeno: 4.7.2017.)

[3] Hu, F. et al.: *A review on cloud computing: Design challenges in architecture and security*, J. Comput. Inf. Technol., vol. 19, no. 1, pp. 25–55, Fakultet elektrotehnike i računalstva, Zagreb, 2011.

[4] Mell, P., T. Grance, T.: *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology*, SAD, 2011.

[5] Bauer, E. and Adams, R.: *Reliability and Availability of Cloud Computing*, p. 352, IEEE, 2012.

[6] Hashemi, S.M., Bardsiri, A.K.: *Cloud Computing Vs . Grid Computing*, vol. 2, no. 5, pp. 188–194, ARPN Journal of Systems and Software, 2012.

[7] Yu, X., Bian, J.: *Reliability Analysis of Cloud Computing Service System*, no. Aice, Joint International Conference on Artificial Intelligence and Computer Engineering, Kina, 2016.

[8] Ray, K.: *Introduction to Service Oriented Architecture*, 2010. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.465.4469&rep=rep1&type=pdf (pristupljeno 4.9.2017.)

[9] Shawish, A., Salama, M.: *Cloud Computing: Paradigms and Technologies*, pp. 39–68, Kairo, 2014.

[10] Stupar, I.: *Ostvarivanje kvalitete usluge u računarstvu u oblaku putem skalabilnosti i sporazuma o razini usluge*, Ericsson Nikola Tesla, Zagreb, 2012.

[11] Malathi, M.: *Cloud Computing Concepts*, pp. 236–239, Department of Computer Science, India 2011.

[12] Gong, C., Liu, J., Zhang, Q., Chen, H., Gong, Z.; *The Characteristics of Cloud Computing*, National University of Defense Technology Changsha, Kina, 2010.

[13] Zhang, Q., Cheng, L., Boutaba, R.: *Cloud computing: state-of-the-art and research challenges*, pp. 7–18, The Brazilian Computer Society, Brazil, 2010.

[14] Rajesh, S., Swapna, S., Reddy, P. S.: *Data as a Service (Daas) in Cloud Computing [Data-As-A-Service in the Age of Data]*, vol. 12, no. 11, Global Journal of Computer Science and Technology, SAD, 2012.

[15] URL: http://cloudscaling.com/blog/cloud-computing/hybrid-clouds-are-half-baked/. (pristupljeno: 11.7.2017.)

[16] Kurze, T., Klems, M., Bermbach, D., Lenk, A., Tai, S., Kunze, M.: *Cloud Federation*, Institute of Applied Informatics and Formal Description Methods, Njemačka, 2013.

[17] Villegas D. et al.: *Cloud federation in a layered service model*, J. Comput. Syst. Sci., vol. 78, no. 5, pp. 1330–1344, Journal of Computer and System Sciences, SAD, 2012.

[18] Gani, A., Anuar, N.B., Shiraz, M., Haque, M. N., Haque, I. T.: *Cloud Service Selection*

Using Multicriteria Decision Analysis, Scientific World Journal, 2014.

[19] Focus Group on Cloud Computing ITU-T: *Requirements and framework architecture of cloud infrastructure*, 2012.

[20] Weinstock, C.B., Goodenough, J.B.: *On System Scalability*, Carnegie Mellon University, SAD, 2006.

[21] Mysore, R. N., et al.: *A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric*, University of California San Diego, 2009.

[22] Azodolmolky, S., Wieder, P., Yahyapour, R., Wissenschaftliche, G.: *Cloud Computing Networking: Challenges and Opportunities for Innovations*, no. July, pp. 54–62, IEEE Communications Magazine, 2013.

[23] Tate, J., Beck, P., Ibarra, H. H., Kumaravel, S., Miklas, L.: *Introduction to Storage Area Networks*, IBM RedBooks, SAD,2016.

[24] CARnet, *Umrežena spremišta podataka*, Zagreb, 2009.

[25] Moghaddam, F. F., Rohani, M. B., Ahmadi, M., Khodadadi, T., Madadipouya, K.: *Cloud computing: Vision, architecture and Characteristics*, Proc. - 2015 6th IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2015, pp. 1–6, 2016.

[26] Friedman, M., Lewis, M., Girola, M.,Tarenzio, A.M.: *IBM Data Center Networking Planning for Virtualization*, IBM RedBooks, 2011.

[27] Van Der Pol, R.: *TRILL and IEEE 802.1aq Overview*, 2012.

[28] Decusatis, C. J. S., Carranza, A.: *Communication within Clouds : Open Standards and Proprietary Protocols for Data Center Networking*, no. September, pp. 26–33, New York City College of Technology, SAD, 2012.

[29] Cisco, The Role of 10 Gigabit Ethernet in Virtualized Environments, pp. 1–9, 2010.

[30] Racherla, S., Erdenberger, S., Rajagopal, H., Ruth, K.: *Storage and Network Convergence Using FCoE and iSCSI*, IBM RedBooks, SAD, 2012.

[31] Cisco: *Data Center Strategy, Architecture and Solutions*, pp. 1–16, 2009. [32] H. Lee, "Virtualization Basics: Understanding Techniques and Fundamentals," 2014.

[33] Saleem, M.: *Cloud Computing Virtualization*, vol. 6, no. 7, pp. 290–292, International Journal of Computer Applications Technology and Research, 2017.

[34] Alshaer, H.: *An overview of network virtualization and cloud network as a service*, pp. 1–30, International Journal Of Network Managemet2015.

[35] URL: https://aws.amazon.com/ec2/faqs/ (pristupljeno: 30.8.2017.)

[36] URL: https://www.geekwire.com/2017/amazon-web-services-secret-weapon-custom-made-hardware-network/." (pristupljeno: 2.9.2017.)

[37] Bhagattjee, B.: *Emergence and Taxonomy of Big Data as a Service*, no. May, Massachusetts Institute of Technology, Cambridge, SAD; 2014.

[38] Amazon Web Services: *Amazon Elastic Compute Cloud User Guide for Linux Instances*, 2017.

[39] Duan, Q.: *Cloud service performance evaluation: status, challenges and opportunities –*

a survey from the system modeling perspective, Digit. Commun. Networks, vol. 3, no. 2, pp. 101–111, 2017.

[40] Wang, G.: *The Impact of Virtualization on Network Performance of Amazon EC2 Data Center*, Proceedings of the 29th conference on Information communications, 2010.

[41] Roloff, E. et al.: *Evaluating High Performance Computing on the Windows Azure*, no. June, Federal University of Rio Grande do Sul, Porto Alegre, Brazil, 2012.

[42] URL: "https://en.wikipedia.org/wiki/Google_Data_Centers." (pristupljeno: 2.9.2017.)

[43] Gandhi, P. V. A., Uni, S.,: *Comparative study of Amazon EC2 and Microsoft Azure cloud architecture*, Int. J. Adv. Netw. Appl., pp. 117–123, 2014.

[44] Firestone, D.: *A Comparison of Public Clouds: Amazon Web Services, Windows Azure and Google App Engine*, pp. 1–3, MIT; SAD, 2011.

[45] Hill, Z., Li, J., Mao, M., Ruiz-alvarez, A., Humphrey, M.: *Early Observations on the Performance of Windows Azure*, Chicago, SAD, 2010.

[46] URL: "http://www.cloudmonix.com/blog/what-is-web-and-worker-role-in-microsoft-azure/." (pristupljeno: 3.8.2017.)

[47] Google: *Google App Engine – the platform for your next great idea*, 2012.

[48] Zahariev, A.: *Google App Engine*, Helsinki University of Technology, Finska, 2009.

[49] Khan, K., Jan, A.: *Evaluating Google App Engine for Enterprise Application Development*, no. June, Blekinge Institute of Technology, Švedska, 2011.

[50] Ghemawat, S., Gobioff, H., Leung, S.: *The Google File System*, Google, New York, SAD, 2003.

[51] Kune, R., Konugurthi, P. K., Agarwal, A.: *The anatomy of big data computing*, pp. 79–105, India, 2016.

[52] URL:"https://en.wikipedia.org/wiki/Application_programming_interface." (pristupljeno: 30.8.2017.)

[53] Kavran, Z., Grgurević, I.: *Predavanja kolegija Računalne mreže*, Sveučilište u Zagrebu, Fakultet prometnih znanosti, Zagreb, 2016.

[54] Mrvelj, Š.: *Predavanja kolegija Tehnologija telekomunikacijskog prometa I*, Sveučilište u Zagrebu, Fakultet prometnih znanosti, Zagreb, 2016.

[55] Periša, M. i Peraković, D.: *Predavanja kolegija Informacijski sustavi mrežnih operatera*, Sveučilište u Zagrebu, Fakultet prometnih znanosti, Zagreb, 2017.

[56] URL: "https://en.wikipedia.org/wiki/Overhead_(computing)." (pristupljeno: 30.8.2017.)

[57] URL: "https://www.geekwire.com/2017/amazon-web-services-secret-weapon-custom-made-hardware-network/." (pristupljeno: 2.9.2017.)

Popis kratica i akronima

AHP	(Analytic Hierarchy Process) analitički hijerarhijski proces
AMIs	(Amazon Machine Images) Amazon slike mašine
API	(Application Programming Interface) aplikacijsko programsko sučelje
ARP	(Address Resolution Protocol) protokol adresnog riješenja
AWS	(Amazon Web Services) Amazon Web usluge
CNA	(Converted Network Adapters) adapteri konvergirane mreže
CPU	(Central Processing Unit) središnja procesna jedinica
CRM	(Customer Relationship Management) upravljanje korisničkim odnosom
DaaS	(Dana as a Service) podatci kao usluga
DAS	(Direct Attached Storage) izravno spojena pohrana
DNS	(Domain Name System) domenski sustav imena
EC2	(Elastic Cloud Computing) elastično računalstvo u oblaku
EOR	(End of Rack) kraj reda
FC	(Fiber Channel)
FCoE	(Fiber Channel over Ethernet)
FICON	(Fiber Connection)
GFS	(Google File System)
HBA	(Host Bus Adapter) sabirnički adapter poslužitelja
HPC	(High Performance Computing) računalstvo visokih performansi
IaaS	(Infrastructure as a Service) infrastruktura kao usluga
iSCI	(internet Small Comuter System Interface) sučelje za mali računalni sustav
IS-IS	(Intermediate System to Intermediate System) posredni sustav prema posrednom sustavu
JBOID	(Just Bunch Of Disks) hrpa diskova

KVM	(Kernel-based Virtual Machine) virtualna mašina temeljena na kernelu
MAC	(Media Access Control) kontrola pristupa mediju
MAN	(Metropolitan Area Network) mreža na području grada
MAUT	(Multiattribute Utility Theory) više-atributna teorija korisnosti
MCDA	(Multicriteria Decision Analysis) višekriterijska analiza odluke
NAS	(Network-attached Storage) umrežena spremišta podataka
NFS	(Network File System) mrežni datotečni sustav
NIST	(National Institute of Standard and Technology) institut za standarde i tehnologiju
OSS	(Operational Supporting System) operacijski sustav podrške
PaaS	(Platform as a Service) platforma kao usluga
PV	(Paravirtualization) para-virtualizacija
RAID	(Redundant Array of Independent Disks) redundantni spremnici za pohranu
SaaS	(Software as a Service) softver kao usluga
SAN	(Storage Area Network) mreža za pohranu
SDN	(Software-defined Networking) softverski temeljeno računalstvo
SLA	(Service Level Agreement) ugovor o razini usluge
SOA	(Service Oriented Architecture) servisno-orijentirana arhitektura
STP	(Spanning Tree Protocol)
TCP	(Transmission Control Protocol)
TOPSIS	(Technique for Order of Preferences by Similarity to Ideal Solution) tehnika za određivanje prednosti preko sličnosti s idealnim rješenjem
TOR	(Top of Rack) na vrhu reda
TRILL	(Transparent Interconnection of Lots of Links) transparentna povezanost puno kanala
VLAN	(Virtual Local Area Network) virtualna lokalna mreža
VM	(Virtual Machine) virtualna mašina

VMM (Virtual Machine Monitor) monitor virtualne mašine

VPC (Virtual Private Cloud) virtualni privatni cloud

VPN (Virtual Private Network) virtualna privatna mreža

VPS (Virtual Private Server) virtualni privatni server

WAN (Wide Area Network) mreža šireg područja

Popis ilustracija

Popis slika

Slika 1. Model dostave Cloud Computinga 9
Slika 2. Model Cloud federacije 14
Slika 3. Referentna arhitektura Cloud Computinga za skalabilnost 16
Slika 4. Radni okvir za cloud infrastrukturu 19
Slika 5. Mrežni model cloud computinga 20
Slika 6. SAN mreža 22
Slika 7. Tipična mrežna arhitektura prospojnika podatkovnog centra 25
Slika 8. Potpuna virtualizacija 31
Slika 9. Paravirtualizacija 32

Popis tablica

Tablica 1. Klasifikacija Cloud Computing usluga 10