

**SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE**

**POSLIJEDIPLOMSKI DOKTORSKI STUDIJ
ELEKTROTEHNIKE I INFORMACIJSKE TEHNOLOGIJE**

KVALIFIKACIJSKI ISPIT

**POHRANA VELIKIH KOLIČINA PODATAKA
NA NOVIM RAČUNALNIM PARADIGMAMA**

Petra Lončar

Split, ožujak 2019.

SADRŽAJ

1. UVOD	1
2. PODATKOVNO INTENZIVNE PARADIGME	3
2.1. Računalstvo u grozdu	3
2.2. Računalni splet	4
2.3. Računalstvo u oblaku	6
2.4. Računalstvo u magli	9
2.5. Rosno računalstvo	12
3. VELIKE KOLIČINE PODATAKA	15
3.1. Značajke velikih količina podataka	15
3.2. Tipovi velikih količina podataka	16
3.2.1. Strukturirani podaci	16
3.2.2. Nestrukturirani podaci	16
3.2.3. Polustrukturirani podaci	17
3.2.4. Metapodaci	17
4. POHRANA VELIKIH KOLIČINA PODATAKA	18
4.1. Koncepti pohrane velikih količina podataka	18
4.1.1. Datotečna pohrana	18
4.1.2. Blok pohrana	19
4.1.3. Objektna pohrana	19
4.2. Tehnologije pohrane velikih količina podataka	20
4.2.1. Diskovne tehnologije za pohranu	20
4.2.2. Memorijske tehnologije za pohranu	24
5. OBRADA VELIKIH KOLIČINA PODATAKA	26
5.1. Hadoop	26
5.2. Spark	29
6. ANALIZA VELIKIH KOLIČINA PODATAKA	32
7. PREGLED LITERATURE I MOGUĆNOSTI ISTRAŽIVANJA	34
7.1. CERN – najveći svjetski znanstveni laboratorij	37
8. ZAKLJUČAK	44
LITERATURA	46
POPIS OZNAKA I KRATICA	52

1. UVOD

Razvoj informacijsko-komunikacijske tehnologije doveo je do nezaustavljivog rasta i gomilanja podataka, a time stvorio i potrebu za pravilnim skladištenjem i obradom velikih količina podataka (engl. *Big Data*). Temeljne značajke velikih količina podataka su veliki obujam, raznolikost i brzina njihovog nastanka. Podaci se generiraju visokom frekvencijom i njihova količina neprestano raste. Razvojem tehnologija i pojavom Internet stvari (engl. *Internet of Things*), evolucijskog koraka u razvoju Interneta, povećao se broj izvora podataka kao što su mobilni telefoni i aplikacije, društvene mreže, digitalna televizija, podaci raznih Internet objekata, senzora i sl. Analiza i pravilna interpretacija koje se odvijaju na najnovijim raspodijeljenim platformama ključne su za velike količine podataka iz kojih se mogu dobiti povratne informacije korisne u područjima industrije, financija, zdravstva, znanstvenih institucija. Primjene i koristi od ove interdisciplinarnе tehnologije podatkovne znanosti su raznolike, a brojni su problemi koje treba uzeti u obzir pri usvajanju pristupa analitici podataka.

Pojam velika količina podataka dolazi iz računalne znanosti i programskog inženjerstva, a opisuje setove podataka koji se povećavaju takvom brzinom da ih je nemoguće obraditi bez superračunala ili računalnih grozdova. Danas se susrećemo s ubrzanim gomilanjem i rastom kompleksnosti podataka iz vanjskog okruženja i unutarnjeg okruženja tvrtki. Velike količine podataka i brzi razvoj tehnologije moraju jednako pratiti i razvijati rješenja za adekvatnom i funkcionalnom pohranom i obradom podataka.

Mooreov zakon [1] kaže da se broj tranzistora koji se po najpovoljnijoj cijeni mogu smjestiti na čip udvostručava otprilike svake dvije godine. Opisuje povećanje broja tranzistora na čipu tokom vremena što je rezultiralo kontinuiranim napretkom tehnoloških rješenja, eksponencijalnim rastom brzine, snage mikroprocesora, a eksponencijalnim padom cijene takvih sustava. Upitna je primjenjivost Mooreovog zakona na velike količine podataka i budući razvoj tehnologije s obzirom da se volumen podataka više nego udvostručuje svakih 18 mjeseci pa će tehnologija premašiti granicu koju nameću fizički zakoni.

Cilj ovog rada je dati sistematski pregled računalnih koncepata i tehnologija, i analizirati mogućnosti upravljanja pohranom velikih količina podataka. Predstaviti će se i postojeća rješenja i identificirati moguća područja budućeg istraživanja. U drugom poglavlju predstaviti će se paradigme raspodijeljenog računalstva, a u trećem poglavlju temeljna definicija velikih količina podataka. U

čtvrtom poglavlju predstaviti će se koncepti i tehnologije za pohranu velikih količina podataka. U petom će se analizirati najpoznatiji okviri za obradu velikih količina podataka, a u šestom poglavlju navesti koraci analize velikih količina podataka. U sedmom poglavlju razmotriti će se trendovi i mogućnosti za pohranu i upravljanje velikim količinama podataka, a u osmom donijeti zaključak rada.

2. PODATKOVNO INTENZIVNE PARADIGME

Fizička infrastruktura zasniva se na raspodijeljenom računalnom modelu što podrazumijeva pohranu podataka na različitim lokacijama koje mogu biti povezane putem mreže, raspodijeljenim datotečnim sustavima i raznim analitičkim alatima i aplikacijama. Fizička infrastruktura mora zadovoljiti kriterije skalabilnosti, otvorenosti, transparentnosti i kvalitete usluge.

Predmet analize ovog poglavlja su paradigme raspodijeljenog računalstva koje su obilježile i unijele revoluciju u računalstvu (Slika 2.4.).

2.1. Računalstvo u grozdu

Računalstvo u grozdu [2] ili klaster računalstvo (engl. *cluster computing*) je sustav nezavisnih, lokalno umreženih računala koji djeluju kooperativno i koordinirano i čine jedinstven računalni resurs. Visok stupanj integracije računala postiže se korištenjem specifične programske podrške. Korištenje raspodijeljene memorije (engl. *distributed memory*) i heterogena arhitektura razlikuju grozd od jedinstvenog višeprosorskog sustava. Računalni grozdovi ovisno o načinu uporabe podržavaju i ostvaruju:

- računalstvo visokih performansi (engl. *High Performance Computing, HPC*)
- računalstvo s visokom propusnošću (engl. *High Throughput Computing, HTC*)
- računalstvo visoke dostupnosti (engl. *High Availability Computing, HAC*)

Za ostvarenje što boljih performansi grozda, pri realizaciji grozdova potrebno je osigurati uravnoteženo opterećenje čvorova koji će dijeliti opterećenje sustava. Zatim, bitno je osigurati da se prilikom pojave greške u radu nekog od čvorova ne zaustavlja obrada na grozdu, usmjeravanjem obrade čvora koji je u kvaru na redundantne čvorove.

Razna znanstvena područja kao što su matematika [3], biologija [4], medicina, računalstvo imaju potrebu za HPC ili HTC infrastrukturom. Visoke performanse računalno-zahtevnih HPC zadataka ostvaruju se njihovim paralelnim izvođenjem i zajedničkim korištenjem raspodijeljene memorije za komunikaciju. Cilj računalstva visokih performansi je brza obrada velike količine podataka. Performanse računalstva visokih performansi su usporedive sa performansama superračunala. Značajke HPC infrastrukture su višejezgreni procesori, veća količina memorije po ugrađenoj jezgri

procesora, velika memorijska propusnost, korištenje brze memorije, osiguravanje istovremenog paralelnog pristupa datotekama, brza i pouzdana pohrana podataka.

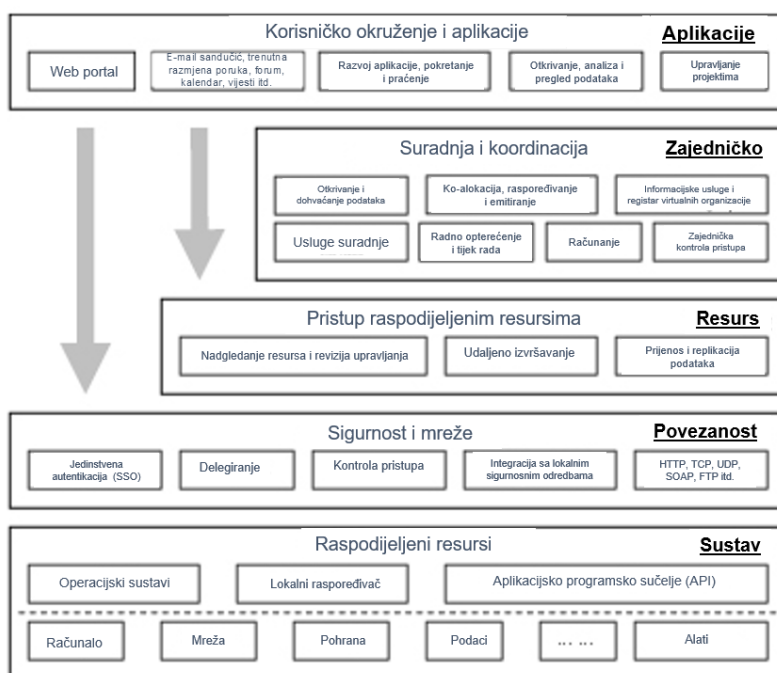
Sustavi računalstva u grozdu koriste se za paralelno programiranje u kojem se računalno zahtjevan program izvršava paralelno na više računala s vlastitom instancom operacijskog sustava. Arhitekturu računalstva u grozdu čine računala, računalna mreža koja treba omogućiti male latencije pri prijenosu velikih količina podataka, operacijski sustav (najprihvatljiviji je Linux) i posrednički sustav (engl. *clusterware*). Posrednički sustav grozda ima zadatak upravljati poslovima, nadgledati rad sustava, omogućiti razmjenu i dijeljenje podataka te pružiti razvojno okruženje za procese. Homogenost čvorova karakteristika je računalstva u grozdu. Jedna od prednosti računalstva u grozdu pred ostalim paradigmama raspodijeljenog računalstva je vlasništvo i upravljanje grozda od strane jednog autoriteta. To omogućava puni pristup sklopovlju i uklanja sve potrebe za udruživanjem. Za postizanje optimalnih performansi, korisnici mogu prilagoditi grozd i aplikacije. Dijeljenje resursa za izvršavanje aplikacija i zadataka nije prisutno unutar grozda. Aplikacija za izvršavanje koristi cjelokupno sklopovlje što omogućava postizanje boljih performansi od primjerice, izvršavanja u oblaku. Nedostaci grozd računalstva su potreba za značajnim ulaganjima i održavanjem, kao i moguća razdoblja nedovoljne iskorištenosti resursa grozda.

2.2. Računalni splet

Po definiciji, računalni splet (engl. *grid computing*) [5], predstavlja skup sklopovske i programske infrastrukture koja omogućava pristup računalnim resursima vrhunskih performansi na pouzdan, dosljedan, univerzalan i povoljan način. Računalni splet omogućava razmjenu i dijeljenje računalne snage i prostora za pohranu podataka. Realizira se umrežavanjem računala i računalnih grozdova preko Interneta. Ideja računalnog spleta je omogućiti jednostavno priključivanje korisnika u tako veliku mrežu udaljenih računala poput priključivanja na električnu mrežu. Od grozda ga razlikuju slabiji stupanj povezanosti računalnih čvorova (radne stanice, grozdovi, poslužitelji) koji se nalaze pod različitim administrativnim domenama, pozicioniranost na različitim geografskim lokacijama i heterogenost (s obzirom na različite sklopovske platforme, arhitekture). Korištenje sredstava u virtualnim organizacijama treba biti sigurno, koordinirano, fleksibilno, kontrolirano i odgovarajućeg stupnja kvalitete usluge (engl. *Quality of Service, QoS*). Cilj računalnog spleta je pružiti visoke performanse i visoku skalabilnost. Računalni i podatkovni splet računala koriste se za pohranu i obradu velike količine podataka, znanstvene simulacije itd.

Osnovni dijelovi slojevite arhitekture računalnog spleta, prikazano na Slika 2.1. su:

- sloj osnovnih sredstava (memorija, procesor, prostor za pohranu, mrežna sredstva, senzori...)
- sloj komunikacijskih protokola
- sloj protokola za sredstva (informacijski i upravljački protokoli za interakciju korisnika s udaljenim sredstvima i uslugama)
- sloj zajedničkih usluga (usluge za raspoređivanje zahtjeva za pristup sredstvima, usluge za nadgledanje rada skupine sredstava, okružja za razvoj logike suradnje sredstava, usluge za obračunavanje i naplatu korištenja sredstava)
- sloj korisničkih aplikacija



Slika 2.1. Arhitektura spleta računala. Prilagođeno prema [6].

Posrednički sloj je virtualizacijski sloj koji pruža osnovne funkcionalnosti kao upravljanje resursima, upravljanje podacima, sigurnosna kontrola, nadgledanje i otkrivanje kvarova i napada. S obzirom na njihovu primarnu namjenu, razlikuju se spletovi za složene proračune (engl. *computational grids*) koji obrađuju CPU zahtjevne poslove kao što su modeliranje i simuliranje složenih znanstvenih eksperimenata, zatim podatkovni spletovi računala (engl. *data grids*) za spremanje i obradu velike količine podataka. Prednosti spleta računala su sposobnost rješavanja složenih i zahtjevnih problema u kratkom roku, jednostavnije korištenje i koordinirano dijeljenje resursa unutar dinamičkih organizacija, bez centralne kontrole, manji ukupni troškovi.

2.3. Računalstvo u oblaku

Računalstvo u oblaku (engl. *cloud computing*) je novija paradigma raspodijeljenog računalstva koja na zahtjev pruža i mrežno dostupnim čini računalne resurse kao što su pohrana, obrada, aplikacije i razne usluge. NIST (engl. *National Institute of Standards and Technology*) je 2010. godine objavio definiciju računalstva u oblaku [7]. Računalstvo u oblaku predstavlja revoluciju u obradi podataka i skalabilnosti koju pruža.

Koncept računalstva u oblaku uključuje postignuća raspodijeljenog računalstva i pohrane, virtualizacije, Internet tehnologija i upravljanja sustavima. Računalstvo u oblaku predstavlja prekretnicu u načinu korištenja tehnologije i optimizaciji korištenja resursa u skladu s promjenjivim potrebama i načinu plaćanja korištenih usluga. Oblaci su troškovno učinkovite platforme za visoko skalabilne pretrage baza podataka, raspodijeljenu pohranu i obradu putem Interneta.

Dostignuća računalstva u oblaku značajna su ne samo za računalnu znanost i industriju već i za niz akademskih disciplina. Računalstvo u oblaku se pojavljuje u literaturi različitih područja uključujući ekonomiju [8], bioinformatiku [9], medicinu [10] te sve veći broj organizacija u telekomunikacijskom i ICT sektoru koji prelazi na neki od oblika poslovanja u oblaku kako bi smanjili operativne troškove i unaprijedili rezultate.

Računalstvo u oblaku omogućava brz pristup fleksibilnim IT resursima putem Interneta onda kada su ta sredstva korisniku potrebna, npr. spremnicima za pohranu, poslužiteljima, bazama podataka i širokom setu aplikacijskih usluga po principu plaćanja po potrošnji (engl. *pay-per-use*). Korisnik nema potrebu za velikim ulaganjima u sklopovlje i njegovo upravljanje. Prednosti ove paradigme su i elastičnost i skalabilnost, širok mrežni pristup preko različitih heterogenih uređaja (mobilni uređaji, prijenosna računala i sl.), mogućnost izolacije preopterećenog poslužitelja ili poslužitelja u kvaru i premještanje zadatka na drugu lokaciju u oblaku, neovisnost o lokaciji resursa koju je moguće definirati na višoj razini apstrakcije (država, podatkovni centar), brzo i fleksibilno pružanje resursa. Sigurnosna pitanja, osjetljivosti podataka u oblaku i pristup podacima od strane neovlaštenih korisnika predstavljaju određeni rizik. Za korisnika usluga u oblaku, nužno je da pružatelj usluga jamči pouzdanost oblaka.

Virtualizacija je jedna od tehnologija koja je temelj razvoja ove paradigme. Ona generalizira fizičku infrastrukturu i čini je komponentom dostupnom i jednostavnom za korištenje i upravljanje. Prednosti

korištenja virtualizacije su: pristup na zahtjev poslužiteljima i resursima za pohranu, ušteda energije, smanjenje potrebnog prostora, racionalno korištenje ljudskih resursa, smanjenje kapitalnih i operativnih troškova, usklađivanje, premještanje i raspodjela opterećenja i izolacija. Usklađivanje više pojedinačnih i heterogenih opterećenja na jednoj fizičkoj platformi omogućuje bolju iskoristivost sustava. Premještanje radnog opterećenja pruža podršku sklopovlju, balansira opterećenje i omogućava oporavak od katastrofe. Provodi se enkapsulacijom stanja operativnog sustava unutar virtualnog stroja (engl. *virtual machine*, *VM*) što omogućava migraciju na druge platforme, nastavak rada ili očuvanje za kasnije pokretanje. Stanje virtualnog stroja uključuje kompletnu sliku diska ili particije, konfiguracijske datoteke i sliku RAM-a.

U virtualiziranim okruženjima hipervizor izvodi apstrakciju sklopovlja na pojedine virtualne strojeve, nadzire, upravlja sustavom u složenom okruženju s više verzija različitih IT resursa i omogućava prisutnost određenih aplikacija na više sustava bez fizičkog mapiranja na svakom od sustava. Hipervizor se može provesti izravno iznad sklopovlja sustava ili iznad operativnog sustava. Hipervizor podržava izvršavanje više virtualnih strojeva, usklađuje njihove aktivnosti i osigurava dosljedan pristup procesoru, memoriji i ulaznim/izlaznim resursima fizičkog stroja. Tehnički različito izvedena i sve češća je virtualizacija kontejnerom (engl. *container virtualization*), gdje se fizičko računalo podijeli u više virtualnih računala od kojih svaki ima vlastiti podskup fizičkih resursa. Kod virtualizacije kontejnerima nema hipervizora, nema virtualiziranog sklopovlja i provodi se na razini operacijskog sustava. Performanse hipervizorske i kontejnerske virtualizacije uspoređene su u [11].

NIST definicijom [7] definirana su tri tipa servisnih modela u oblaku (Slika 2.2.):

- **platforma kao usluga** (engl. *Platform as a Service*, *PaaS*) - korisnik koristi alate koje pruža davatelj usluga u oblaku. Alati mogu uključivati programske jezike i alate za razvoj aplikacija, razvoj sučelja, razvoj baze podataka, pohranu i testiranje. Korisnik usluga u oblaku ne upravlja infrastrukturom u oblaku, no može upravljati aplikacijama i konfiguracijama okruženja u kojem se ugrađuju (npr. Amazon Web Services, Google App Engine itd.)
- **softver kao usluga** (engl. *Software as a Service*, *SaaS*) - poslužitelj u oblaku omogućuje korisniku pristup softverskim aplikacijama koje se pokreću na infrastrukturi u oblaku putem Interneta. Jedna od prednosti SaaS usluga je i jednostavnija nadogradnja i ažuriranje usluga za sve korisnike. Google je jedan od primjera pružatelja usluga u oblaku (npr. Google Apps - Docs, Gmail, Sheets, Drive...)
- **infrastruktura kao usluga** (engl. *Infrastructure as a Service*, *IaaS*) - usluga kojom se korisnicima oblaka omogućava korištenje IT infrastrukture, tj. računalnih, mrežnih i resursa

za pohranu i drugih osnovnih računalnih resursa gdje korisnik usluga u oblaku može implementirati i pokrenuti željenu aplikaciju. Korisnik usluge u oblaku može kontrolirati operativni sustav, implementirati aplikacije i imati ograničenu kontrolu nad mrežnim uređajima. IaaS nudi najveći stupanj kontrole, u usporedbi s PaaS i SaaS. Amazon je pružatelj usluga koji nudi IaaS i omogućuje korisniku da učita svoj operacijski sustav i aplikaciju na Amazon infrastrukturu (Amazon Elastic Compute Cloud, Google Compute Engine...).



Slika 2.2. Usluge i arhitekture računalstva u oblaku. Prilagođeno prema [12].

Uz osnovne usluge koje su definirane računalstvom u oblaku razvijaju se nove usluge kojima se u marketinške svrhe dodaje sufiks "as a Service (aaS)" zbirnog naziva XaaS (engl. *Everything as a Service*) [13]. Neke od tih usluga koje nisu podržane službenom NIST definicijom su: Storage as a Service, Database as a Service, Analytics as a Service, Network as a Service, Backup as a Service, Management as a Service, Firewall as a Service, Voice as a Service, Mobility as a Service i mnoge druge.

Izvedbeni modeli računalstva u oblaku su: privatni oblak (engl. *private cloud*) kojeg koristi jedna organizacija i njim upravlja organizacija ili 3rd-party, javni oblak (engl. *public cloud*) omogućava javnoj publici otvoreno korištenje infrastrukture pružatelja usluga u oblaku, zajednica oblaka (engl. *community cloud*) pruža usluge u oblaku grupi organizacija zajedničkih interesa i hibridni oblak (engl. *hybrid cloud*) koji nastaje kombinacijom i povezivanjem dva ili više prethodno navedenih modela oblaka.

Pregledom osnovnih značajki računalstva u oblaku, kao njegove prednosti treba istaknuti fleksibilnost i skalabilnost, niže troškove, organizacijsku agilnost, demokratizaciju korištenja resursa, a kao nedostatke privatnost podataka, dostupnost, sigurnost, upravljanje podacima, odgovornost pružatelja usluga.

2.4. Računalstvo u magli

Računalstvo u magli (engl. *fog computing, fogging*) je nova paradigma raspodijeljenog računalstva koja objedinjuje načela računalstva u oblaku i rastuću industriju Interneta stvari (engl. *Internet of Things, IoT*). Cisco je [14] uveo računalstvo u magli kao proširenje računalstva u oblaku prema rubu mreže (engl. *edge computing*), bliže krajnjim korisnicima. Koncept rubnog računalstva nastao je kao odgovor na naglo eksponencijalno povećanje i složenost podataka, uređaja i umrežavanje sustava. Procjena je da će do 2020. godine mrežno povezano biti 50 milijardi „stvari“ [15].

Računalstvo u magli podrazumijeva korištenje virtualizacije i dijeljenje resursa u visoko skalabilnim podatkovnim centrima i mikro podatkovnim centrima u blizini malih uređaja (pametnih uređaja, usmjernika i drugih mrežnih uređaja). Osnovna načela računalstva u oblaku su iz centra mreže izmještena na rub mreže da krajnjim korisnicima pruže podatke, obradu, pohranu i usluge. Suradnja s računalstvom u oblaku koji nudi pohranu na zahtjev i skalabilnost i dalje je potrebna. Važno je odrediti koje poslove obraditi na računalnim resursima u blizini, a koje isporučiti u podatkovne centre računalstva u oblaku, kao i optimizirati alokaciju resursa za takve zadatke. Kašnjenja, problemi performansi sustava, centraliziranost čvorova, sigurnost podataka, potrošnja energije [20] kao i nužna povezanost Internetom i osiguravanje dostatne propusnosti neki su od nedostataka računalstva u oblaku koje računalstvo u magli može prevladati. Približavanjem obrade i analize izvoru podataka smanjuje se i latencija obrade, smanjuje se potrošnja energije, rasterećuje se oblak, a osigurana je skalabilnost izvedbe. Senzori i drugi IoT uređaji i aplikacije pametnih gradova, autonomnih vozila, robota, zdravstvenih aplikacija i ostalih usluga koje utječu na kvalitetu života, svakodnevno generiraju velike količine osjetljivih podataka na koje je potrebno pravovremeno (u milisekundama) i pouzdano reagirati. Takvi uređaji koji koriste komunikacije stroja sa strojem (engl. *machine-to-machine, M2M*) ili interakciju čovjeka i stroja, uz unaprijeđenost obrade mogu iskoristiti prednosti pojave ovog koncepta i obrade bliže izvoru podataka. Prednosti računalstva u magli nad drugim pristupima opisuju se kroz termin SCALE [18] (engl. *Security, Cognition, Agility, Latency, Efficiency*) koji označava da se transakcije izvode uz osiguranu zaštitu, usmjerenost na klijente za osiguravanje autonomije, agilnost, obradu u stvarnom vremenu i kontrola sustava i dinamičko

udruživanje nekorištenih resursa uređaja krajnjih korisnika. Računalstvo u magli olakšava upravljanje i programiranje mrežnih, računalnih i usluga pohrane između podatkovnih centara i krajnjih uređaja, podržava mobilnost korisnika, heterogenost resursa i raspodijeljenu analizu podataka kako bi udovoljili zahtjevima široko raspodijeljenih aplikacija koje zahtijevaju nisku latenciju. Osim u IoT, računalstvo u magli će važnu ulogu imati i u ostvarivanju mobilnih mreža i usluga pete generacije (5G) [16], postizanju boljih performansi web stranica [17] i slično.

Ulogu fizičkih uređaja na kojima se odvija računalstvo u magli, tj. čvorova u magli mogu obnašati kontroleri, usmjernici, poslužitelji, mrežni preklopnici, nadzorne kamere, i drugi uređaji koji imaju mogućnost obrade, pohrane i umrežavanja Bluetooth, WiFi ili Ethernet tehnologijom.

U oblak se uz pomoć IP protokola šalju odabrani podaci koje je potrebno dugotrajno arhivirati i detaljnije analizirati. Budući da čvor računalstva u magli može biti povezan s drugim čvorovima u magli ili s oblakom, komunikacijske razine (Slika 2.3.) koje treba razmatrati i analizirati su oblak-magla, magla-magla, magla-rosa (engl. *cloud-fog*, *fog-fog* i *fog-dew*) i to s obzirom na dijeljenje pohrane, računalnih i mrežnih resursa, upravljačkih mehanizama.



Slika 2.3. Komunikacijske razine računalstva u magli. Prilagođeno prema [18].

Dakle, glavne značajke računalstva u magli su:

- niska latencija kao rezultat korištenja čvorova u magli za smanjenje udaljenosti između IoT uređaja i oblaka
- poznata lokacija čvorova u magli na kojima se mogu izvoditi određene aplikacije što također smanjuje latenciju i skraćuje vrijeme izvršavanja zadataka

- obrada osjetljivih podataka unutar jezgre mreže
- široka zemljopisna rasprostranjenost IoT uređaja
- visoka skalabilnost senzorskih mreža koje komuniciraju s čvorovima u magli i na osnovi njihove dostupnosti dodjeljuju im poslove
- podrška za mobilnost uređaja
- interakcija u stvarnom vremenu što je osobito važno za vremenski osjetljive aplikacije koje traže brzu reakciju
- heterogenost IoT uređaja za koja nisu definirana sučelja i standardi

Open Fog Consortium [19] je udruženje akademskih institucija i visokotehnoških tvrtki koje daje podršku razvoju, istraživanju i standardizaciji računalstva u magli. Sklopovska infrastruktura platforme računalstva u magli ima zadaću osigurati podršku i zaštitu za unutarnje komponente, osigurati usklađenost sa standardnim propisima, zaštititi ih od okolišnih čimbenika kojima će biti izloženi (temperatura, vlažnost, vibracije), upravljati hlađenjem unutarnjih komponenti, udovoljiti zahtjevima za sigurnost ljudi i stvari, otpornost na fizički napad ili krađu, prihvatljivu veličinu, potrošnju energije i težinu komponenata, podršku za modularnost konfigurabilnih komponenti (CPU, RAM i SSD komponente, različite konfiguracije pohrane, akceleratori za osiguravanje dodatne računalne propusnosti kao što su FPGA i GPU, konfigurabilan I/O za podršku različitim sučeljima) i mogućnost proširenja, servisiranje platforme. Platforme u magli umjerene veličine u rubnim ili *on-premise* platformama mogu se zamijeniti razdjelnom pločom (engl. *backplane*) i podržati modularnost na razini čvora. Najveće platforme nalikovat će plošnim poslužiteljima visokog kapaciteta podržavajući mnogo modula, sofisticirane višeučitelne CPU farme, nizove GPUova, tisuće I/O linkova i pouzdanu pohranu veličine PB s podrškom za enkripciju i autentikaciju.

Najnižu razinu u IoT čine senzori, aktuatori i druga programska podrška ili sklopovski uređaji kojih može biti i nekoliko tisuća, a povezani su žičnim ili bežičnim protokolima ovisno o lokaciji i svrsi čvora, pokrivenosti, mobilnosti, volumenu i brzini prijenosa podataka. Mreža treba osigurati skalabilnost, fleksibilnost, QoS i dostupnost koju traži proces, zadovoljavajući tražene udaljenosti i kapacitete. Svaki čvor treba komunicirati putem mreže koja ovisno o vremenskoj osjetljivosti aplikacije treba imati podršku za vremenski osjetljivo umrežavanje (TSN, engl. *Time Sensitive Networking*), npr. u Industriji 4.0 TSN je ključan, u automatizaciji, robotskim okruženjima. Sve to dovodi do potrebe za kontrolom, nadzorom i upravljanjem mrežom, resursima, sigurnošću, prilagodbom promjenjivim uvjetima okružja, tj. *out-of-band* (OOB) mrežnim upravljanjem sustava.

Virtualizacija sklopovlja koristi se za implementaciju platforme u magli, omogućava entitetima da dijele isti fizički sustav, a pridonosi i sigurnosti sustava. Tri su programske razine čvora u magli: razina upravljanja čvorom i programska razdjelna ploča, podrška aplikacijama i razina aplikacijskih usluga. Programska razdjelna ploča pokreće programe na čvoru i osigurava komunikaciju među čvorovima, a uključuje operacijski sustav, upravljačke programe i datoteke nadogradnje (engl. *firmware*), komunikacijske usluge, datotečni sustav, virtualizaciju programske podrške i podršku za izolaciju temeljenu na operacijskom sustavu za pokrenutu programsku podršku i aplikacijske mikro usluge. Ova razina mora osigurati pouzdanost podataka i integritet usluge i tako zaštititi komunikaciju između stvari i magle, magle i magle i magle i oblaka. Svaki čvor treba imati sljedeće mogućnosti: upravljanje konfiguracijom, operacijsko upravljanje, upravljanje sigurnošću, upravljanje kapacitetom i upravljanje dostupnošću u kritičnim situacijama.

Razina podrške aplikacijama ovisi o nižim razinama, a uključuje ovjeru, ažuriranje i osiguravanje programskog sučelja za podršku aplikacijama i mikro uslugama, aplikacijske poslužitelje, VM-ine, biblioteke programskih jezika, kontejnere i izvršno okruženje za aplikacije i mikro usluge. Također, uključuje podršku za aplikacije temeljene na događajima i porukama, sigurnosne usluge uključujući usluge enkripcije, upravljanje podacima, pohranom i trajnom pohranom, mogućnost transformacije podataka i pohrane, alate za analizu. Najviša razina, razina aplikacijskih usluga, ključnih usluga koje prikupljaju podatke i čine ih dostupnim drugim uslugama i sustavima, usluge podrške za registraciju, čišćenje podataka, registraciju usluge, praćenje zapisa, analitičke usluge, integracijske usluge koje čvorovima izvan magle dopuštaju da se predbilježe za podatke od interesa koji su generirani u magli i definiraju gdje, kako i u kojem formatu im treba dostaviti podatke.

2.5. Rosno računalstvo

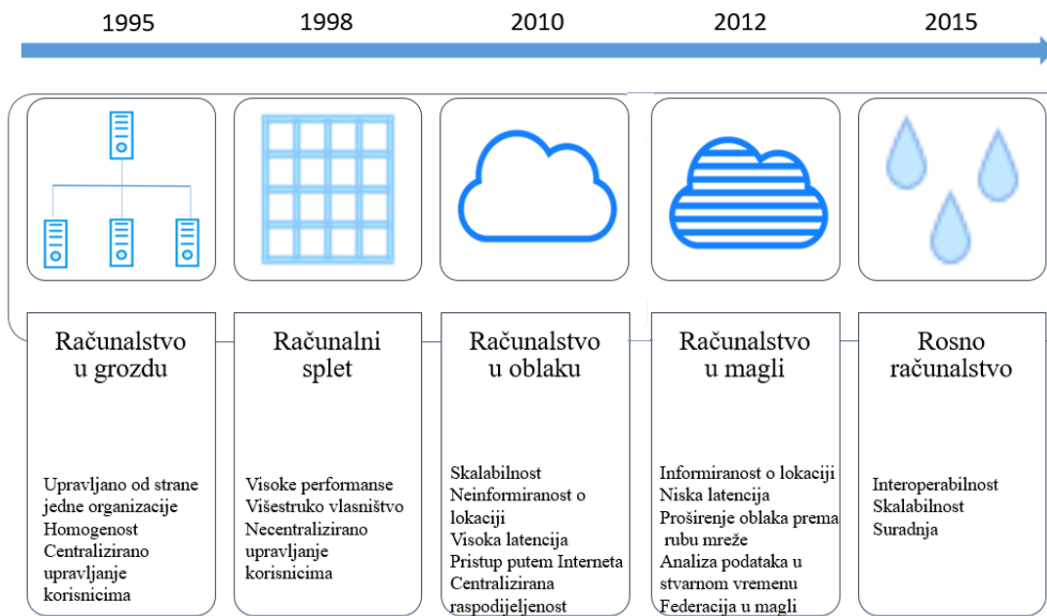
Daljnji razvoj računalstva doveo je do još jedne programsko-sklopovske paradigme računalstva koja za cilj ima u potpunosti iskoristiti *on-premise* računala, nazvane rosno računalstvo ili *dew computing* [21]. Osnova rosnog računalstva je hijerarhijska oblak-rosa arhitektura opisana u [22] čije je ključno svojstvo suradnja mikro i makro usluga, tj. usluga u rosi i oblaku. Naime, uređaj u rosi (osobno računalo, tablet, pametni mobitel, poslužitelj i sl.) s odgovarajućom programskom podrškom pruža funkcionalnosti neovisno o uslugama u oblaku, a s njima i surađuje.

Dok računalstvo u magli uključuje senzore, usmjernike i automatizirane IoT uređaje, rosno računalstvo nastoji ostvariti puni potencijal *on-premise* resursa, tj. resursa koji se ne koriste u

računalstvu u oblaku. Iako rosno računalstvo nije u prvom redu namijenjeno IoT-u, ono također može ponuditi rješenja za primjenu u IoT-u i služi kao posrednik između magle koji uključuje mnogobrojne automatizacijske uređaje i oblaka koji koristi velike poslužitelje i baze podataka. DC surađuje s uslugama u oblaku i kontrolira uređaje u magli. Ideja rosnog računalstva je maksimalno iskoristiti resurse prije nego se obrada prenese poslužitelju u oblaku, reducirati složenost i unaprijediti produktivnost skalabilnog raspodijeljenog računalstva. Usluge u rosi koriste manje uređaje koji mogu odraditi kompresiju manje slike, manje količine podataka, zvuka. Heterogeni uređaji u rosi su programibilni i rekonfigurabilni kako bi mogli učinkovito obavljati složene zadatke koji koriste različite alate i aplikacije.

Hijerarhijska arhitektura paradigmi rosa, magla i oblak ostvaruje skalabilnost vertikalno, na svim razinama i proširuje model klijent-poslužitelj. Uz poslužitelj u oblaku koji pruža usluge u oblaku, uvodi se i poslužitelj u rosi [23], web poslužitelj koji se nalazi na lokalnom korisničkom računalu. Poslužitelj u rosi ima zadatak samo jednom klijentu korisniku pružiti usluge koje nudi i poslužitelj u oblaku, i sinkronizirati bazu podataka poslužitelja u rosi sa onom na poslužitelju u oblaku. Kapacitetom je manji od poslužitelja u oblaku, pohranjuje samo podatke korisnika, osjetljiv je na viruse i greške, jednostavno ga je ponovno instalirati uz pomoć kopije podataka u oblaku i moguće mu je pristupiti neovisno o Internet vezi jer se izvodi na lokalnom računalu. Dvije značajke računalstva u rosi su neovisnost i suradnja. Upravo jedan od nedostataka računalstva u magli i oblaku je nužnost uspostavljene Internet veze u svakom trenutku te dostupnost poslužitelja u oblaku. U takvim uvjetima, korisnik ne može pristupiti podacima i teško može sinkronizirati računalne zadatke. Neovisnost rosnog računalstva leži u činjenici da aplikacije u rosi nisu u potpunosti *online*, a suradnja označava da one moraju koristiti usluge računalstva u oblaku i s njima automatski izmjenjivati informacije tijekom izvođenja. Primjer aplikacija u rosi su Dropbox, OneDrive i Google Drive Offline. One nude usluge pohrane korisniku u svakom trenutku, a sinkroniziraju se sa uslugama u oblaku i pripadaju kategoriji *Storage in Dew (STiD)*. Potencijalna primjena rosnog računalstva je i u *Web in Dew (WiD)* kojim je moguće pretraživanje bez Internet veze uz podršku posebnog Web poslužitelja na *on-premise* računalu koji bi lokalno podržavao dublicirane web adrese koje bi se sinkronizirale sa web stranicama u oblaku pri uspostavljenju Internet veze [21]. Ostale usluge u rosnom računalstvu su: *Database in Dew (DBiD)*, *Software in Dew (SiD)*, *Platform in Dew (PiD)*, *Infrastructure as Dew (IaD)*, *Data in Dew (DiD)* [21].

Izazovi rosnog računalstva su na području ključnih elemenata ove nove paradigme, razvoj sklopovske opreme, mreža, operacijski sustav, baze podataka, poslužitelji i pretraživači.



Slika 2.4. Paradigme raspodijeljenog računalstva. Prilagođeno prema [90].

3. VELIKE KOLIČINE PODATAKA

Tehnologija velikih količina podataka je trenutno među najvažnijim tehnologijama istraživanja u znanosti, biologiji, inženjerstvu, medicini, poduzetništvu, astronomiji, društvenim znanostima i ostalim područjima u kojima kao rezultat provedenih eksperimenata i simulacija nastaju velike količine podataka. Ova tehnologija ima mogućnost značajno promijeniti način na koji organizacije koriste informacije kako bi unaprijedile iskustvo korisnika i transformirati poslovne modele pretvarajući informacije u znanje. Za implementaciju rješenja tehnologije velikih količina podataka potrebna je odgovarajuća infrastruktura koja podržava skalabilnost, distribuciju i upravljanje podacima.

3.1. Značajke velikih količina podataka

Koncept znanosti o podacima ima dugu povijest, a u posljednje vrijeme je posebno popularna zbog rastuće upotrebe računalstva u oblaku i koncepta Interneta stvari (IoT) u stvaranju pametnog okruženja, kao i velikog korištenja društvenih mreža (Facebook, Twitter i dr.) i pojave četvrte industrijske revolucije. Veliki podaci su opisani 5V karakteristikama (engl. *volume*, *velocity*, *value*, *veracity*, *variety*), odnosno volumen, brzina, vrijednost, vjerodostojnost i raznolikost podataka (Slika 3.1.). Podatke velikog volumena, iz raznovrsnih raspodijeljenih izvora i u različitim formatima koji pristižu velikom brzinom i koje treba obraditi u realnom vremenu, zahtjevno je pratiti i statistički predvidjeti te se njihova autentičnost i vrijednost mogu značajno razlikovati ovisno o načinu rukovanja njima. Kompleksni i heterogeni podaci veličine jednog terabajta (1 TB) i više smatraju se velikim količinama podataka. Tako velike količine strukturiranih podataka (organiziranih u formi baza ili tablica), polustrukturiranih (e-mail) i nestrukturiranih (multimedijski sadržaj, fotografije, zvuk, video, GPS podaci, sadržaj kojeg generira korisnik) podataka traže skalabilnost odgovarajućih resursa za njihov prihvata, pohranu, pretraživanje, dijeljenje, pretvorbu, upravljanje, analizu i vizualizaciju. Upravo analiza i pravilna interpretacija daju značaj podacima. Velike količine podataka predstavljaju prekretnicu u razvoju društva i ekonomije temeljene na podacima. Korištenje velikih količina podataka iziskuje nove tehnologije za rukovanje njima i njihovu analizu.

5 značajki velikih količina podataka



Slika 3.1. Značajke velikih količina podataka, 5V. Prilagođeno prema [24].

3.2. Tipovi velikih količina podataka

Podaci koji treba obraditi i analizom dobiti vrijedne informacije mogu biti strojno generirani (logovi web stranica, senzorski podaci i sl.) ili nastati kao rezultat interakcija čovjeka i sustava. Generirani podaci nisu istog tipa i ne dolaze u istoj formi. Osnovni tipovi u kojima se pojavljuju podaci su: strukturirani, polustrukturirani i nestrukturirani podaci. Od značaja su i metapodaci.

3.2.1. Strukturirani podaci

Strukturirani podaci udovoljavaju formalno određenom predlošku ili podatkovnom modelu i pohranjuju se u tabličnoj strukturi. Model definira pravila izgleda podataka i operacije koje je moguće izvoditi nad njima. Procjenjuje se da strukturirani podaci čine oko 20% ukupne količine podataka. Strukturirani podaci prikazuju odnose među različitim entitetima, a najčešće ih generiraju poslovne aplikacije i upravljački informacijski sustavi. Pohranjuju se u relacijskim bazama podataka (RDBMS, engl. *Relational Database Management System*). Primjeri strukturiranih podataka su stringovi, bankovne transakcije, Excel dokumenti, RFID senzorski podaci, GPS podaci. Za njihovo upravljanje koriste se postojeći alati i baze podataka (zasnovane na SQL jeziku) i uglavnom ne postavljaju posebne zahtjeve za obradu i pohranu.

3.2.2. Nestrukturirani podaci

Podaci koji nemaju definiran format i nisu u skladu s određenim podatkovnim modelom ili podatkovnim predloškom nazivaju se nestrukturirani podaci. Nestrukturirani podaci stvaraju se većim

brzinama i čine oko 80% ukupne količine podataka. Dolaze u formi tekstualnih ili binarnih podataka čiji je sadržaj nestrukturiran. Tekstualne datoteke sadržavaju web objave ili objave na društvenim mrežama, dok binarne datoteke sadrže slike, audio ili video sadržaj (npr. satelitske snimke, fotografije, meteorološki podaci, podaci u fizici visokih energija). Postojeća tehnologija ne omogućava više od pohrane i ručne analize nestrukturiranih podataka. Za pohranu nestrukturiranih podataka koriste se nerelacijske baze podataka kao što je NoSQL (engl. *Not-only Structrured Query Language*).

3.2.3. Polustrukturirani podaci

Polustrukturirani podaci imaju definiranu strukturu i dosljednost koja je hijerarhijska ili je zasnovana na strukturi grafa, dakle nije nerelacijska. Polustrukturirani podaci sadrže značajke strukturiranih i nestrukturiranih podataka. Ovakvi podaci se pohranjuju u datotekama koje sadrže tekst. Primjer polustrukturiranih formi podataka su XML i JSON datoteke, kao proračunske tablice, senzorski podaci, dokumenti elektronske razmjene podataka (engl. *EDI, Electronic Data Interchange*). Budući da posjeduju određen stupanj strukture, jednostavnije ih je obraditi od podataka koji ne posjeduju definiranu strukturu.

3.2.4. Metapodaci

Metapodaci pružaju informacije o značajkama i strukturi setova podataka. Ovi strojno generirani podaci su ključna komponenta pri integraciji podataka jer nose dodatne informacije o podacima. Značajni su za obradu, pohranu i analizu velikih količina podataka. Primjer metapodataka su XML tagovi i atributi koji sadrže informacije o autoru, datumu kreiranja datoteke, veličini i rezoluciji fotografije, razini privatnosti i sl.

4. POHRANA VELIKIH KOLIČINA PODATAKA

Prikupljanje, pohrana, pretraživanje, dijeljenje, pretvorba, analiza i vizualizacija korištenjem tradicionalnih tehnologija i alata postaju sve složeniji za postojeće računalne sustave. Razvoj novih tehnologija i metodologija ima za cilj pronaći rješenja na izazov podataka velikih dimenzija. Obrada i upravljanje pohranom ključni su za performanse podatkovno intenzivnih računalnih sustava. Sustavi se razlikuju u korištenim podatkovnim modelima, sustavskoj arhitekturi, korištenim pristupima za postizanje skalabilnosti i visoke dostupnosti te korisničkim sučeljima.

Velike količine podataka imaju važnu ulogu u sustavima za pohranu podatkovno intenzivnog računalstva. Volumen, brzina i raznolikost podataka su izazov za pohranu i analizu velikih podataka. Do sada korišteni načini pohrane kao što su tvrdi diskovi (HDD) i RAID (engl. *Redundant Array of Independent Disks*) mehanizam nisu pouzdani za pohranu podataka volumena reda nekoliko petabajta (PB). Jedan od zahtjeva za pohranu velikih podataka je i visoka skalabilnost sredstava u pristupu horizontalnog skaliranja (engl. *scale-out*), dodavanjem dodatnih čvorova u sustav, koja treba biti brzo ostvarena uz zadovoljavajući trošak i osiguravanje podrške za strukturirane i nestrukturirane podatke kojima bi se pristupalo velikim brzinama uz osiguranu privatnost. U ovom poglavlju predstaviti će se koncepti i tehnologije pohrane velikih količina podataka.

4.1. Koncepti pohrane velikih količina podataka

Razvojem različitih tehnologija pojavljuju se raznovrsni podaci koji se stalno razmjenjuju putem mreže. Kako bi takvi podaci bili svakodnevno dostupni raznim korisnicima sustava, dolazi do primjene raznovrsnih spremišta podataka. Postoje tri osnovna koncepta za pohranu podataka, a to su objektna (engl. *object storage*), datotečna (engl. *file storage*) i blok pohrana (engl. *block storage*) (Slika 4.1.).

4.1.1. Datotečna pohrana

Pohrana temeljena na datotekama koristi datotečni sustav za označavanje lokacije na uređaju za pohranu na kojoj su pohranjeni podaci. Ovaj koncept pohrane koristi se na izravnim i umreženim sustavima za pohranu podataka, a brine o organizaciji i prikazu podataka korisnicima. Podaci su raspoređeni na poslužitelju u istom formatu kao što ih vide klijenti. Korisniku je na taj način omogućeno da traži datoteku prema jedinstvenom identifikatoru (naziv, lokacija, URL) koji se

komunicira sa sustavom za pohranu pomoću specifičnih protokola za prijenos podataka. U hijerarhijskoj strukturi, spremanje datoteka je iznad razine pohrane u formi blokova. To korisnicima omogućuje pristup podacima kao datotekama, ali istovremeno ograničava pristup blokovima pod tim datotekama. Dijeljenim datotečnim sustavima poput NFS-a (engl. *Network File System*) poslužitelji mogu pristupiti preko IP mreže. Pristup se može kontrolirati na razini datoteke, direktorija i izvoza putem korisničkih i grupnih dopuštenja. Pohrana datoteka može se koristiti za pohranu podataka koje koristi više korisnika i strojeva, za lokalno dijeljenje podataka (NAS, engl. *Network Attached Storage*), za baze podataka te virtualne strojeve i kontejnere.

4.1.2. Blok pohrana

Pohrana podataka u blokovima fiksne duljine predstavlja stariji oblik pohrane podataka u aplikacijama koje zahtijevaju postojane visoke performanse. Pohranjeni volumen kontrolira se operacijskim sustavom vanjskog poslužitelja, a može se promatrati kao neovisni disk uređaj. Većina aplikacija koristi ovaj tip pohrane. Pohrana strukturiranih podataka u blokovima odvija se na magnetnim vrpčama, NAND flash memoriji, tvrdim diskovima, optičkim diskovima, RAID sustavima, email poslužiteljima (Microsoft Exchange), virtualnim strojevima (VMware), ali i neke vrste pohrane u oblaku (Amazon Elastic Block Storage, Rackspace Cloud Block Storage, Google Persistent Disks i dr.). iSCSI (engl. *Internet Small Computer System Interface*), FCoE (engl. *Fibre Channel over Ethernet*) i drugi standardi temeljeni na IP protokolu koriste se za povezivanje objekata za pohranu podataka i upravljanje spremištima na velikoj udaljenosti.

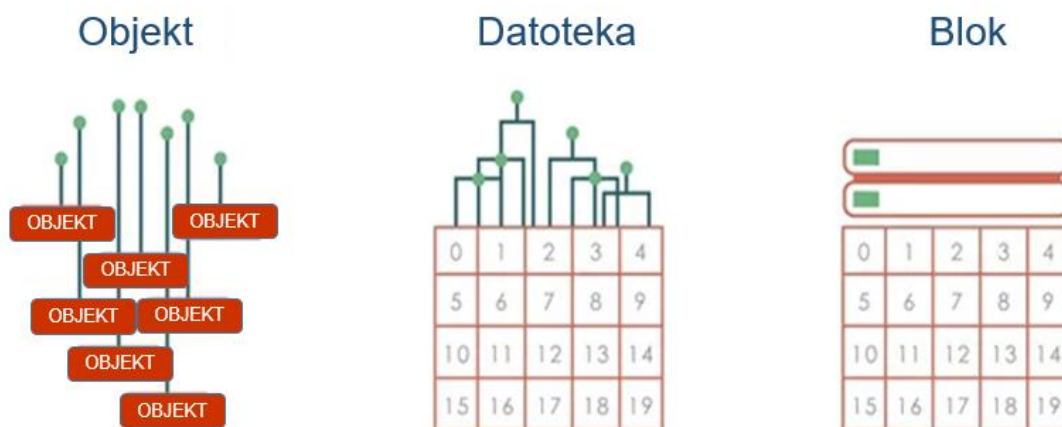
4.1.3. Objektna pohrana

Koncept pohrane temeljen na objektima je obećavajući koncept pohrane podataka za superračunalstvo i pohranu podataka reda PB i više. Najnovija forma pohrane pruža bolju skalabilnost, elastičnost, fleksibilnost, lokalnu transparentnost i trajnost od često korištenih datotečnih sustava. Podaci se pohranjuju u izolirane kontejnere nazvane objektima nad kojima su podržane dvije osnovne operacije: PUT (stvara novi objekt i popunjava ga podacima) i GET (dohvaća sadržaj objekta temeljem jedinstvenog identifikatora objekta). U objekt je moguće samo jednom zapisivati.

Podaci mogu fizički biti pohranjeni na lokalnom poslužitelju ili na geografski miljama udaljenim poslužiteljima. Računalstvo u oblaku pohranu velikih setova nestrukturiranih podataka (glazba, fotografije, video zapisi) zasniva na objektima razdvajanjem sadržaja od indeksa i omogućava

spajanje više datoteka u objekt. Metapodaci pripojeni sadržaju objekta pružaju kontekst o bajtovima sadržanim unutar objekta. Ostali primjeri primjene objektne pohrane uključuju velike količine podataka, web aplikacije, arhiviranje sigurnosnih kopija i sl. Podaci pohranjeni kao objekti mogu biti replicirani u različitim podatkovnim centrima, a pristup objektima putem REST API protokola je vrlo brz i na velikoj skali.

Amazon (Amazon S3), Microsoft (Microsoft Azure), Google (Google Cloud Storage), Rackspace (Rackspace Cloud Files) primjeri su pružatelja usluga pohrane podataka temeljene na objektima.



Slika 4.1. Koncepti pohrane podataka. Prilagođeno prema [25].

4.2. Tehnologije pohrane velikih količina podataka

Idealan sustav za pohranu velikih količina podataka trebao bi osigurati sigurnu pohranu neograničenih količina strukturiranih i nestrukturiranih podataka te podržati različite podatkovne modele. Podjelu sustava za pohranu je moguće izvršiti na one koji koriste disk ili koji koriste memoriju grozda. Takve sustave je jednostavno proširiti i prilagoditi rastućim potrebama za resursima uvođenjem novih čvorova. Tehnološki trendovi sve više se usmjeravaju tehnologijama pohrane unutar memorije (engl. *in-memory*).

4.2.1. Diskovne tehnologije za pohranu

Tradicionalno korišteni sustavi, raspodijeljeni datotečni sustavi ili baze podataka, za dugotrajnu pohranu podataka koriste pristupačne tvrde diskove (engl. *on-disk storage*).

4.2.1.1. RDBMS baze podataka

Glavna namjena podatkovnih baza je obrada velikih količina podataka kojima korisnici mogu pristupiti, definirati ih i ažurirati uz pomoć programske podrške. Programski sustavi za upravljanje relacijskom bazom podataka RDBMS (engl. *Relational Database Management System*) podržavaju pravila relacijskog modela i ACID (atomarnost, dosljednost, izolacija i trajnost, engl. *Atomicity, Consistency, Isolation, Durability*) pravila ograničavajući se na jedno računalo. Skaliranje ovih sustava izvodi se okomito što njihovu izvedbu čini skupljima i neadekvatnima za dugotrajnu pohranu podataka. Podaci se trebaju držati određenog predloška, tj. odgovarati pohrani strukturiranih podataka među kojima postoji povezanost. Podaci se pohranjuju u tablice koje čine stupci i redovi. Osim toga, treba osigurati i dosljednost podataka što je moguć problem kod kontrole pristupa velikoj količini podataka. Zbog postojanja latencije nisu dobar izbor za pohranu velikih količina podataka koji pristižu učestalo velikim brzinama. Popularne relacijske baze podataka su Oracle MySQL [26], Microsoft SQL Server [27] i dr.

4.2.1.2. NoSQL baze podataka

Priroda i struktura podataka koji se stvaraju u današnjim aplikacijama traži nove tehnologije pohrane. Nerelacijske baze podataka predstavljaju važnu tehnologiju za pohranu nestrukturiranih, polustrukturiranih i nepredvidivih podataka jer postojeće funkcionalne i pouzdane relacijske baze podataka koje koriste predvidive, strukturirane podatke unaprijed određenih svojstava ne odgovaraju njihovim potrebama i nije predviđeno da rade nad velikim brojem raspodijeljenih računala. NoSQL (engl. *Not Only SQL*) uspješno odgovaraju na skalabilnost, složenost podatkovnih struktura i performanse raspodijeljenog sustava.

NoSQL sustavi pohrane podataka su u sve većoj upotrebi jer ne zahtijevaju fiksne tablične prikaze, ali i ne zadovoljavaju u potpunosti tradicionalna ACID svojstva. NoSQL sustave veže se uz BASE svojstva (engl. *Basically Available, Soft state, Eventually consistent*), što znači da je većina podataka dostupna veći dio vremena, stanje sustava promjenjivo, a sustav postaje postojan s vremenom. Uz NoSQL baze podataka često se veže i CAP teorem (engl. *Consistency, Availability, Partition tolerance*) [28] koji kaže kako je nemoguće istovremeno zadovoljiti dosljednost, dostupnost i toleranciju na ispad djela raspodijeljenog sustava. Prilikom razvoja sustava za upravljanje bazama podataka moguće je ostvariti samo dva od navedena tri svojstva raspodijeljenih sustava. Dok se tradicionalni sustavi za upravljanje bazama podataka odlučuju na dosljednost i dostupnost, NoSQL

sustavi uvijek ostvaruju toleranciju na podijeljenost kako bi se zadržala jednostavna skalabilnost, a na uštrb dosljednosti (PA) ili dostupnosti (PC).

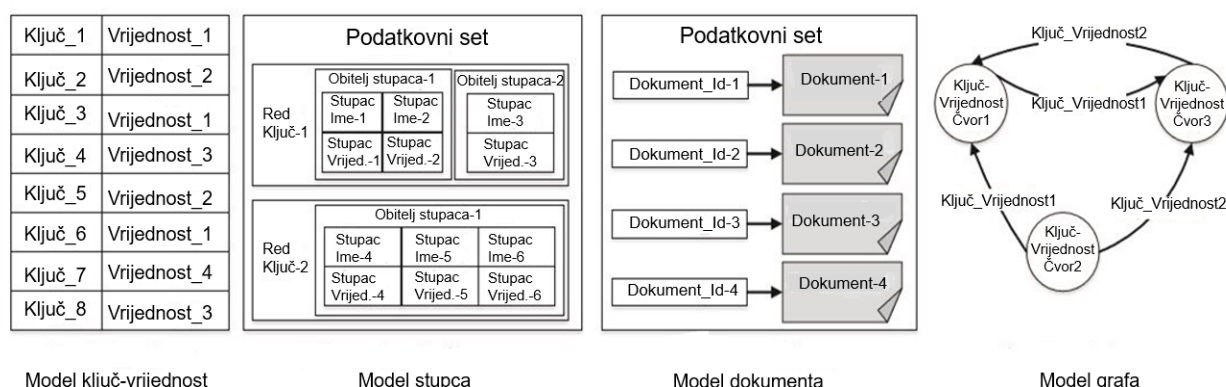
Ove podatkovne baze koriste različite modele pohrane podataka koji ih razlikuju (Slika 4.2.):

- **model ključ-vrijednost** (engl. *key-value store*) - jednostavan, skalabilan model baze zasnovan na parovima podataka ključ-vrijednost koji mogu biti različitih tipova i strukturiranosti. Sve operacije (GET, PUT, DELETE) nad podacima moguće je izvoditi na temelju jedinstvenog ključa. Primjeri industrijskih baza zasnovanih na ovom modelu su Oracle NoSQL Database [29], Dynamo Paper [30], Riak [31] i dr.
- **model stupca** (engl. *column store*) - koriste uređaji koji pohranu podataka organiziraju u stupcima. Svaki stupac je smješten u posebnoj datoteci ili segmentu na disku te može sadržavati proizvoljan broj povezanih stupaca (super-stupac). Prednost stupčanih sustava nad relacijskim bazama podataka, koje se sastoje od redova i stupaca, je u mogućnosti pristupa pojedinačnim stupcima. Brzo učitavanje i pristup stupcima u memoriju, analiziranje, izvršavanje upita i I/O operacija je znatno učinkovitije. Stupčane baze podataka omogućuju i veliku kompresiju podataka. Tehničke karakteristike stupčane baze čine prihvatljivima za obradu velikih količina podataka koji zahtijevaju čitanje/pisanje u stvarnom vremenu. Ove baze su prihvatljive i za rad s podacima koji imaju tabličnu strukturu u kojoj svaki redak sadrži veći broj povezanih stupaca osiguravajući učinkovitu pohranu alocirajući prostor samo za retke koji sadrže stupce. Primjeri stupčanih NoSQL baza su Apache baze otvorenog koda Cassandra [32] i HBase [33], Google Bigtable [34].
- **model grafa** (engl. *graph databases*) - podaci se pohranjuju u strukturi grafa, a veze među čvorovima čine ih prikladnima za pohranu povezanih podataka kao što su podaci s društvenih mreža, ali i mnogi drugi sustavi. Upiti se izvršavaju praćenjem veza među čvorovima nad određenim dijelom grafa koji zadovoljava kriterije definirane u upitu. Budući da performanse sustava nisu ovisne o ukupnom broju čvorova i veza koje se mogu pohraniti u bazi podataka, a ni o vremenu odgovora, upiti se izvršavaju jednakom brzinom koja ovisi o vremenu potrebnom za pronalaženje početnog čvora. Model grafa sa svojstvima je implementiran u graf bazama podataka s podržanim metodama za kreiranje, čitanje, ažuriranje i brisanje (CRUD, engl. *Create, Read, Update, Delete*), uz podržane ACID principe transakcija. Novi podaci mogu se dodati bez velikog utjecaja na postojeći graf. Graf baze nisu prikladne za uporabu u aplikacijama koje zahtijevaju često ažuriranje većeg broja čvorova odjednom i raspodijeljenost resursa. Najpoznatiji sustavi za

upravljanje bazama podataka ovog modela su Neo4J [35], InfiniteGraph [36], OrientDB [37] itd.

- **model dokumenta** (engl. *document databases*) - pohranjuju podatke kao parove ključ-vrijednost, gdje je vrijednost u strukturiranoj ili polustrukturiranoj formi dokumenta (XML, JSON i sl.). Dokumentima se može pristupiti korištenjem ključa, a moguće je i vršiti API upite nad dinamičkom strukturom dokumenta i njegovog sadržaja. Ovakvi sustavi podržavaju replikaciju, pohranu velikih datoteka, transformaciju dokumenata, raspodjelu podataka na više poslužitelja i postizanje boljih performansi, ugnježđivanje dokumenata. MongoDB [38], Terrastore [39] i CouchDB [40] primjeri su datotečnih NoSQL baza podataka.

Pri upravljanju podacima raspodijeljene baze podataka trebaju osigurati mogućnost da se na svaki zahtjev klijenta odgovori s ažuriranim podacima (dosljednost), da poslužitelj primljeni zahtjev obradi i na njega odgovori (dostupnost) i da sustav nastavi s radom u uvjetima kada se dogodila greška u komunikaciji među poslužiteljima (otpornost particije). Za upite se koriste ne-relacijski jezici (ne SQL), primjerice XQuery [41], SPARQL [42].



Slika 4.2. Modeli NoSQL baza. Prilagođeno prema [43].

4.2.1.3. NewSQL baze podataka

Moderna forma relacijskih baza podataka koje pružaju skalabilnost, tolerantnost na pogreške i brze operacije čitanja/pisanja kao i NoSQL baze [43], ali i transakcijske operacije i ACID svojstva relacijskih sustava baza podataka. NewSQL baze podataka koriste SQL relacijski jezik za izvršavanje operacija nad podacima. Pogodne su za korištenje u bankovnim sustavima i analitici u realnom vremenu. Primjeri su Google Spanner [44], NuoDB [45], Apache Ignite [46] i dr.

4.2.1.4. Raspodijeljeni datotečni sustavi

Postojeći datotečni sustavi koji upravljaju dugoročnom pohranom podataka u raspodijeljenim sustavima (DFS, engl. *Distributed File Systems*), prilagođeni su pohrani malih datoteka, dedupliciranju podataka mrežno dijeljenih datoteka i balansiranju opterećenja. Podržavaju mobilnost korisnika i paralelni pristup podacima. Prvi raspodijeljeni datotečni sustav orijentiran velikim podacima nastao je 2003., Google File System. Od 2004. godine razvija se HDFS [47] (*Hadoop Distributed File System*), visoko propusni raspodijeljeni datotečni sustav otvorenog koda koji je osnova Hadoop [48] infrastrukture za skladištenje i obradu velikih količina nestrukturiranih podataka. Primjeri DFS-a uključuju i CephFS [49], MooseFS [50] i GlusterFS [51].

4.2.2. Memorijske tehnologije za pohranu

Osim na disku, podatke je moguće u potpunosti pohraniti u radnoj memoriji računala ili računalnog grozda (engl. *in-memory storage*). Svi pristupi i upiti koji se vrše nad podacima odvijaju se u memoriji s minimalnim uplitanjem diska. Time se postiže veliko ubrzanje obrade informacija prebacivanjem velike količine podataka u brzu memoriju. Smanjenje cijena i povećanje kapaciteta radne memorije dovelo je do razvoja i širokog raspona primjena memorijskih uređaja za pohranu. Podrška za masovnu paralelnu obradu omogućuje horizontalno skaliranje dodavanjem dodatnih čvorova u grozd što je značajno i financijski jer je moguće jednostavno povećati kapacitet grozda i performanse dodavanjem dodatnih poslužitelja. Unatoč tome, skaliranje memorijskih sustava je skuplje od skaliranja sustava temeljenih na disku (zbog ograničenog mjesta za ugradnju memorije). Učinkovitost uređaja se postiže pohranom i kompresijom podataka u stupce što rezultira smanjenim brojem ulaznih i izlaznih operacija prilikom čitanja i zapisivanja podataka (čak 80 puta brže nego na uređajima s pohranom na disku). Nedostatak ovih uređaja proizlazi upravo iz njihove glavne prednosti, korištenja RAM memorije. Svi privremeni podaci bit će izgubljeni ukoliko dođe do pada računalnog sustava. Memorijski uređaji ispunjavaju tri karakteristike ACID modela, atomarnost, dosljednost i izoliranost, ali ne i trajnost. Kao rješenje ovog nedostatka, dodatne kopije podataka mogu se pohraniti na medij za trajnu pohranu. Podržani su podaci s i bez definirane strukture.

Memorijski uređaji prikladni su za pohranu podataka: koji pristižu velikim brzinama i koje je potrebno analizirati u realnom vremenu, potrebna je kontinuirana analiza, različiti procesi se koriste istim setom podataka, zahtijeva se niska latencija. Nisu prikladni za podatke koje treba serijski obraditi i dugotrajno sačuvati. Razlikuju se dva načina implementacije: memorijske baze podataka

(IMDB, engl. *in-memory data grids*) i memorijski spletovi podataka (IMDG, engl. *in-memory database*). Razlikuje ih način na koji su podaci pohranjeni u memoriji.

4.2.2.1. Memorijski podatkovni spletovi

IMDG u memoriji pohranjuju polustrukturirane i nestrukturirane podatke kao parove ključ-vrijednost kojima se pristupa pomoću API-ja. Podržan je i SQL. Sinkronizacija među čvorovima odvija se redovito, a osigurana je i dosljednost izvršavanja naredbi transakcije. Pristup nerelacijskim podacima koji se pohranjuju kao objekti brži je od onog u IMDB. Skalabilnost se postiže replikacijom podataka na barem jedan dodatni čvor koje se koristi u slučaju kvara jednog čvora, ponovno obnavljajući podatke iz postojećih replika. IMDG se uglavnom koristi za analitiku u stvarnom vremenu zbog svojstva aktivnih upita koji koristi filtere i u slučaju promjena obavještava klijente. IMDG se može implementirati u oblaku kao medij za fleksibilnu i visoko skalabilnu pohranu. U implementaciji sustava, često se ostvaruju s diskovnim uređajima za pohranu kroz pristupe: *write-through*, *read-through*, *write-behind* i *refresh-ahead* [52]. Primjeri IMDG su: Oracle Coherence [53], Infinispan [54] i Hazelcast [55].

4.2.2.2. Memorijske baze podataka

IMDB osiguravaju pohranu strukturiranih, polustrukturiranih i nestrukturiranih podataka kroz relacijske i nerelacijske modele baza uz korištenje RAM memorije. IMDB ne zahtijevaju diskovne uređaje za sigurnosnu pohranu podataka te ih je stoga jednostavnije postaviti. Zamjena RDBMS relacijskim memorijskim bazama ne postavlja dodatne zahtjeve, s obzirom da i IMDB pruža podršku za SQL, za razliku od NoSQL IMDB. Zbog podrške raspodijeljenim upitima i ACID transakcijama kroz grozd, skalabilnost relacijskih IMDB je lošija od IMDG. Prikladne su za scenarije u kojima je postojeće rješenje zasnovano na diskovnom modelu potrebno proširiti ili zamijeniti memorijskim rješenjem i tako osigurati podršku u realnom vremenu.

Primjer IMDB baze je Exasol [56], iznimno brza memorijska analitička baza koja koristi paralelnu obradu i komunikacijske tehnike visokih performansi za povećanje iskoristivosti sklopovlja bez ograničenja volumena podataka. Učinkovitost postiže pohranom i kompresijom podataka u stupcima i smanjenom broju ulaznih i izlaznih operacija prilikom čitanja i zapisivanja podataka. Osigurava pohranu strukturiranih, polustrukturiranih/nestrukturiranih podataka kroz relacijske i nerelacijske modele baza. Ostali primjeri su ArangoDB [57], Redis [58], Apache Ignite [59], VoltDB [60] i dr.

5. OBRADA VELIKIH KOLIČINA PODATAKA

Obrada velikih količina podataka odvija se raspodijeljeno i paralelno nad manjim particijama seta, na lokacijama gdje je set pohranjen. Simultanim i konkurentnim izvođenjem obrade nad manjim setovima podataka smanjuje se vrijeme izvođenja i povećava učinkovitost podjelom poslova u raspodijeljenom sustavu. Velike količine podataka, zbog svoje prirode i načina korištenja, često zahtijevaju obradu u realnom vremenu.

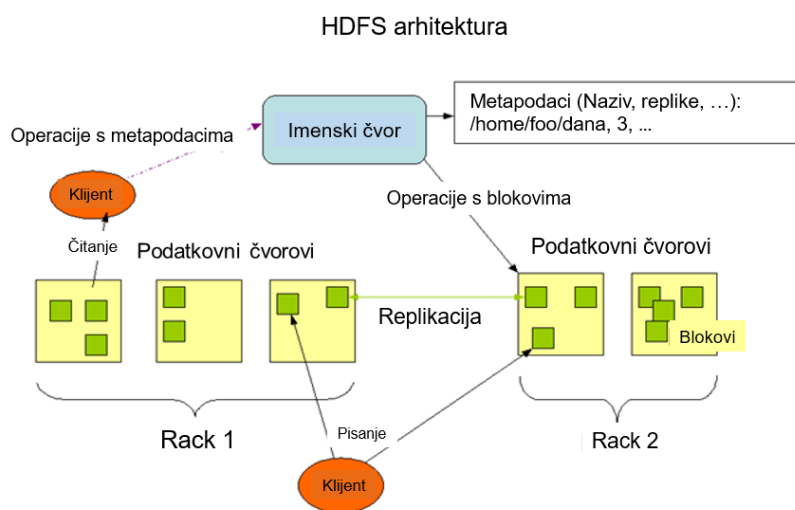
MapReduce [61] paradigma i platforme otvorenog koda Hadoop i Spark postigli su veliku popularnost i važnost kod obrade, stvaranja i analize setova velike količine podataka.

5.1. Hadoop

Apache razvija i održava Hadoop, a definira ga kao programska podrška otvorenog koda za pouzdano i skalabilno raspodijeljeno upravljanje velikim podatkovnim skupovima u računalnim grozdovima. Upravljanje podacima uključuje raspodijeljeno spremanje podataka i očuvanje integriteta spremljenih podataka i pružanje programskog sučelja za upravljanje podacima na jednostavan način. Hadoop omogućuje horizontalnu skalabilnost, od jednog računala do više tisuća računala, gdje svaki nudi svoj lokalni diskovni prostor i mogućnosti obrade podataka. Jedna od najbitnijih komponenti koja omogućuje skaliranje i obradu podataka je Hadoop raspodijeljeni datotečni sustav HDFS (engl. *Hadoop Distributed File System*). HDFS podržava tradicionalnu hijerarhijsku organizaciju datoteka. Omogućava korisnicima sustava upravljanje direktorijima, stvaranje direktorija te spremanje datoteka unutar direktorija. Važne komponente Hadoopa su i upravitelj resursima YARN (engl. *Yet Another Resource Negotiator*) te programski model MapReduce.

HDFS - Hadoop raspodijeljeni datotečni sustav je komponenta sustava zadužena za spremanje podataka. Ova tehnologija namijenjena je računalima prosječnih performansi koje je po potrebi jednostavno skalirati horizontalno i koji imaju zadaću osigurati dugotrajnu pohranu. Datoteke rastavlja na blokove konfigurabilne veličine te ih sprema na više različitih lokacija na pouzdan i siguran način, čime se rješava problem ograničenog diskovnog prostora jednog tvrdog diska. Na taj način datoteka se može spremati na više različitih diskova u više različitih čvorova unutar grozda. Obrada datoteke moguća je na svakom od čvorova na kojem se nalaze njene komponente. Kako bi se ubrzao pristup podacima, datotečni sustav nastoji pronaći najbliže računalo za paralelnu obradu i bolje iskorištavanje dostupnih računalnih resursa. Slika 5.1. prikazuje HDFS master/slave arhitekturu

koju čini imenski čvor (engl. *Namenode*) i podatkovni radni čvorovi (engl. *Datanode*). Imenski čvor je glavni čvor koji upravlja pristupom klijenata datotekama imenskog područja, a radi sa metapodacima. TaskTracker se nalazi na imenskom čvoru, a odgovoran je za izvršavanje zadataka ili podnošenje izvješća o napretku JobTrackeru. JobTrackeri se nalaze na radnim čvorovima i odgovorni su za izvršavanje zadataka primljenih od TaskTrackera. Podatkovni radni čvorovi pohranjuju i poslužuju datoteke koje su podijeljene u podatkovne blokove. Višestruke replikacije blokova pohranjuju se na podatkovnim radnim čvorovima. Podatkovni radni čvorovi zaduženi su za čitanje i pisanje podataka u datotečni sustav po uputama primljenim od imenskog čvora.



Slika 5.1. HDFS arhitektura. Prilagođeno prema [47].

HDFS je napisan Java programskim jezikom. Koriste ga mnoge IT organizacije, uključujući Intel, IBM i Yahoo kao tehnologiju za pohranu velikih količina podataka. HDFS nije najbolje rješenje za aplikacije koje rade u realnom vremenu, s velikim količinama malih datoteka i podataka koje je potrebno često ažurirati.

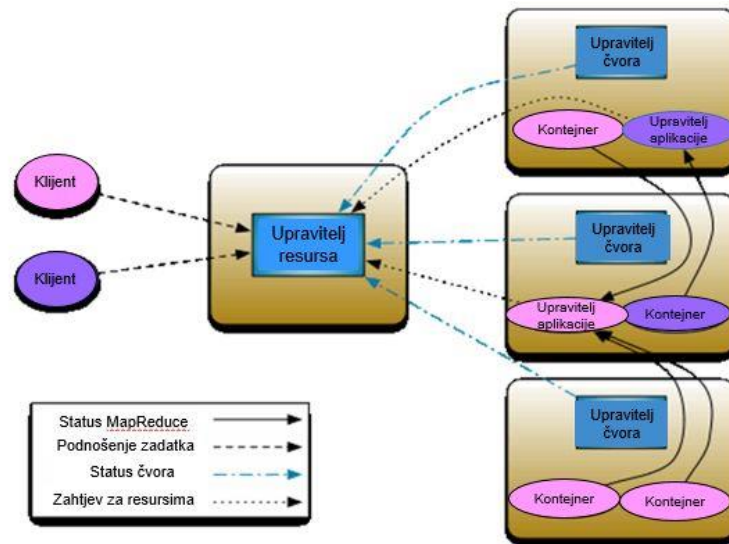
MapReduce - Hadoopov programski model MapReduce [48] omogućava paralelnu, asinkronu obradu velikih količina podataka na grozdu računala. MapReduce paradigma obradu izvodi nad hrpom podataka strukturiranih u parove <ključ, vrijednost>. MapReduce paradigma sastoji se od dvije funkcije, Map i Reduce. Mapper je funkcija zadužena za sortiranje i filtriranje (engl. *shuffle and sort*) ulaznih podataka koje razvrstava prema definiranom ključu. Izlaz je lista parova, koju čine ključ i vrijednosti koje mu pripadaju ([ključ, (v1, v2, ...)]). Map funkcija se u pravilu izvodi na računalima

na kojima je pohranjena kopija podataka za obradu, a njen rezultat se privremeno pohranjuje na lokalnom disku. Slijedi je funkcija koja se naziva Reducer, a zadatak joj je da obrađuje i prikuplja mapirane podatke. Kao rezultat vraća listu parova gdje se svaki par sastoji od ključa i vrijednosti koja mu pripada. Vrijednost se dobiva sumiranjem liste podataka (prebrojavanje podataka, traženje srednje vrijednosti, traženje maksimalne i minimalne vrijednosti, ukupne sume) dobivenih iz prethodne procedure. Svaki podatkovni čvor može zasebno mapirati podatke koji se po zajedničkom ključu dodjeljuju jednom čvoru, a zatim se nad njima može obaviti i procedura Reducer. Upravo to omogućuje obradu velikih količina podataka i skaliranje dodavanjem novih čvorova u grozd. Rezultati MapReduce obrade pohranjuju se u HDFS-u.

Glavni čvor dodjeljuje ulazne podatke radnim čvorovima, osigurava izvođenje funkcija definiranim redoslijedom i upravlja komunikacijom radnih čvorova.

Apache Spark i Apache Storm koriste MapReduce paradigmu u obradi velike količine podataka.

YARN - YARN [63] omogućava zakazivanje poslova i upravljanje resursima dostupnima u grozdu računala. Zadaća YARN-a je alociranje sistemskih resursa za izvođenje aplikacija i određivanje kada i na kojem čvoru raspodijeljenog sustava će se izvršiti poslovi. Glavne komponente YARN-a (Slika 5.2.) su upravitelj resursa (engl. *ResourceManager*) i upravitelj aplikacije (engl. *ApplicationMaster*). Važna komponenta je i upravitelj čvorom (engl. *NodeManager*) koji je instaliran na svakom čvoru i služi za promatranje korištenja resursa i objavljivanje statusa i događaja upravitelju resursa. Upravitelja resursa čine Scheduler i ApplicationsManager. Scheduler je komponenta koja brine samo o alokaciji resursa za aplikacije koje se izvršavaju, ne vodeći računa o njihovom statusu. ApplicationsManager upravlja aplikacijama pisanim za Hadoop ispitujući stanje resursa i odlukama o redoslijedu izvršavanja određenih poslova. Ukoliko dođe do pogreške u radu upravitelja aplikacije, ApplicationMaster će ga resetirati. ApplicationMaster je kreiran za svaku aplikaciju i pregovara o dostupnim resursima i poslovima koji treba obaviti sa upraviteljem čvora, prati status kontejnera resursa dodijeljenih individualnoj aplikaciji.

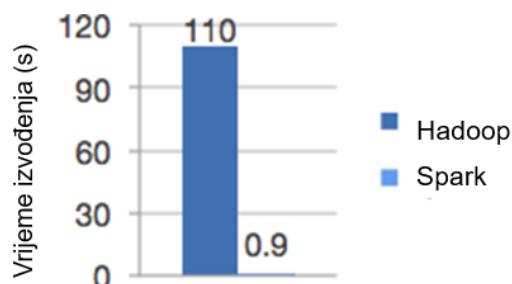


Slika 5.2. Apache YARN. Prilagođeno prema [63].

5.2. Spark

Drugi najpoznatiji i najkorišteniji okvir za obradu velikih količina podataka na grozdovima je Apache Spark [62]. Spark je računalna platforma otvorenog koda za brzu i skalabilnu obradu podataka na grozdu računala. Podržava razne module za različite obrade podataka i raznolik skup API-ja za rad u Javi, Pythonu, R i Scali. Može se izvoditi samostalno, na Hadoopu YARN-u, Apache Mesosu, Kubernetes ili računalnom oblaku.

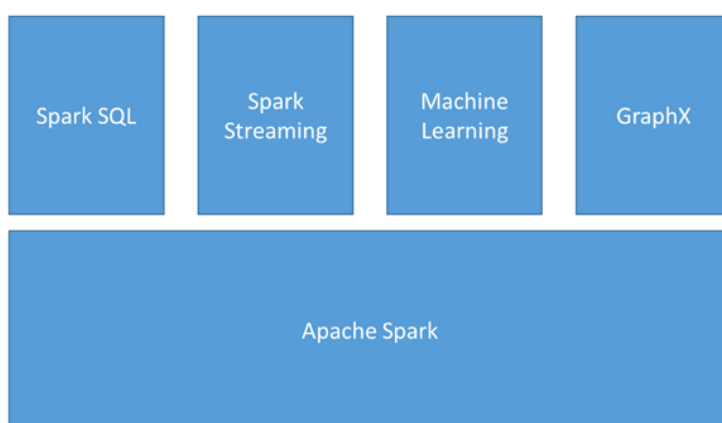
Nema vlastiti sustav za pohranu podataka već radi sa HDFS-om ili nekim drugim sustavom (HBase, Cassandra, Hive, S3 itd.). Spark u jednom koraku čita podatke, obrađuje ih te zapisuje rezultate. Podržava obradu podataka u radnoj memoriji grozda računala čime postiže visoke performanse, što ga čini stotinu puta bržim od Hadoop MapReducea (Slika 5.3.).



Slika 5.3. Usporedba performansi Hadoopa i Sparka. Prilagođeno prema [62].

Ključni moduli Sparka su (Slika 5.4.):

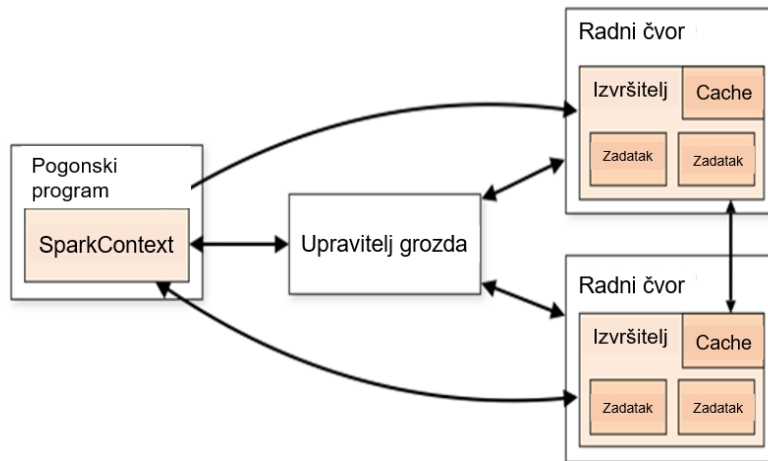
- Spark Core - osnova je Apache Spark arhitekture, pruža podršku za upravljanje poslovanjem i resursima
- Spark SQL - pruža podršku za rad sa strukturiranim podacima i SQL upitima
- Spark Streaming - sposoban za obradu velike količine podataka u stvarnom vremenu
- Machine Learning, MLib - biblioteka je za skalabilno izvođenje algoritama strojnog učenja
- GraphX - pruža podršku za rad s velikim grafovima



Slika 5.4. Apache Spark. Preuzeto iz [62].

Osnovni tipovi podataka u Sparku su RDD (engl. *Resilient Distributed Dataset*) i DataFrame (okvir podataka). RDD je raspodijeljena kolekcija objekata particionirana po grozdovima računala, otporna na pogreške, nepromjenjiva i s mogućnošću paralelnog pristupa. DataFrame je nastao od RDD-a, a predstavlja raspodijeljenu kolekciju podataka organiziranih u imenovane stupce.

Na Slika 5.5. prikazano je izvođenje aplikacije na Sparku. Pokrenut pogonski program (engl. *driver*) je proces u kojem se izvodi glavna metoda aplikacije. Pogonski program je odgovoran za podjelu posla na zadatke, koordiniranje i raspoređivanje poslova na radne čvorove (engl. *worker node*) prema prostornoj raspoređenosti. Radni čvorovi izvode dodijeljene zadatke i u memoriji pohranjuju podatke potrebne za izvođenje aplikacije.



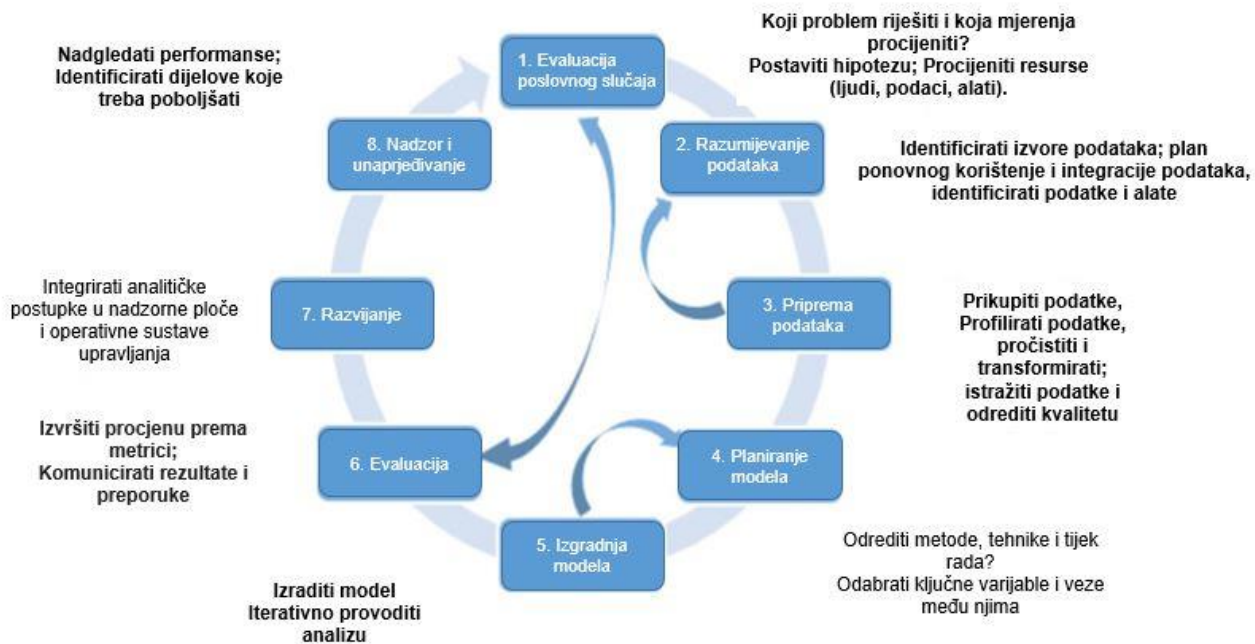
Slika 5.5. Izvođenje aplikacije na Apache Sparku. Prilagođeno prema [64].

6. ANALIZA VELIKIH KOLIČINA PODATAKA

Analiza velikih količina podataka razlikuje se od uobičajene podatkovne analize prvenstveno zbog volumena, brzine i raznolikosti podataka koje je potrebno kako bi se adresirali različiti zahtjevi za analizu podataka. Stručnjaci koji sudjeluju u ciklusu analize podataka trebaju imati posebna znanja i vještine koje uključuju tehnike rudarenja podataka. Odluke temeljene na analitici podataka smatraju se boljim odlukama od onih koje se ne temelje na činjenicama, već na slobodnoj interpretaciji stručnjaka. Korištenjem analitičkih platformi i tehnika, velike količine podataka mogu poboljšati performanse organizacija i organizacijskih jedinica. Razlikuje se nekoliko vrsta analize: kvalitativna, kvantitativna, statistička, vizualna, semantička, rudarenje podacima, strojno učenje. Umjetna inteligencija i strojno učenje imat će velik utjecaj na razvoj usluga iz područja analitike velikih količina podataka, kao što su izrada odgovarajuće strukture podataka, njihovo upravljanje i optimizacija procesa.

Analiza velikih količina podataka provodi se kroz sljedeće korake [65] (Slika 6.1.):

1. Evaluaciju poslovnog slučaja - definiranje potreba i ciljeva analize velikih količina podataka
2. Identifikaciju podataka - identificiranje vlastitih ili vanjskih izvora setova podataka za analizu
3. Pripremu podataka - dohvaćanje podataka iz prethodno definiranih izvora i automatsko uklanjanje (filtriranje) podataka koji nisu od važnosti za analizu te pridruživanje metapodataka, uklanjanje redundantnosti, upotpunjavanje seta podacima koji trenutno nedostaju i potvrđivanje točnosti i kvalitete seta, izdvajanje različitih formata podataka i pretvaranje u format prihvatljiv za daljnju analizu
4. Planiranje modela - odabir metoda, odabir ključnih varijabli i određivanje veza među njima
5. Izradu modela - provođenje iterativnog postupka analize različite složenosti (od jednostavnih upita do složenih statističkih i matematičkih modela) prema modelu
6. Evaluaciju rezultata analize - procjena prema metrici, komuniciranje rezultata i preporuka
7. Razvijanje - integriranje postupaka analize u nadzorne ploče i operativne sustave upravljanja podacima
8. Korištenje rezultata analize - donošenje odluka i primjena rezultata analize za unaprjeđenje i optimizaciju sustava velikih količina podataka



Slika 6.1. Analiza velikih količina podataka. Prilagođeno prema [65].

7. PREGLED LITERATURE I MOGUĆNOSTI ISTRAŽIVANJA

Tehnologije za obradu velikih količina podataka napreduju velikom brzinom. S rastućim trendovima na području velike količine podataka, organizacije trebaju stvoriti uvjete u kojima će analitičari i znanstvenici moći eksperimentirati podacima. Ljudski resursi, ulaganja i kreativne ideje temeljne su sastavnice razvoja područja velikih količina podataka.

U domeni HPC prikupljaju se ogromne količine podataka iz složenih znanstvenih istraživanja kao što su modeliranje klime [66], fizika elementarnih čestica [67], i sl., ali i mnogih industrija, poslovnih [68], biomedicinskih i zdravstvenih primjena [69], [70], društvenih mreža [71], upravljanja elektroenergetskim mrežama [72] i drugih područja koji rade sa setovima podataka reda petabajta. Ostvarivanje znanstvenih postignuća i stvaranje inovativnih proizvoda i usluga nezamislivi su bez modeliranja, simuliranja, i obrade podataka dobivenih u eksperimentima ili podataka opažачkih instrumenata. Tehnologije velikih količina podataka podrazumijevaju korištenje visoko skalabilne raspodijeljene infrastrukture za paralelnu obradu, pohranu, prijenos, pristup udaljenim resursima za zajednički rad velikog broja geografski udaljenih istraživača. Velike organizacije koje dominiraju u radu s velikim količinama podataka poput Googlea, Amazona, Microsofta i Facebooka problem pohrane podataka rješavaju raspodijeljenom arhitekturom računala postižući tako uravnoteženo opterećenje, visoke performanse, visoku dostupnost podataka te mehanizam upravljanja greškama ili pada sustava za pravovremeni pristup podacima.

Pohrana podataka temelj je obrade velikih količina podataka i izdvajanja relevantnih informacija. Tradicionalni HPC sustavi odvajaju računalne resurse od pohrane, koristeći paralelne datotečne sustave, dok moderna superračunala imaju lokalne uređaje za pohranu koji podržavaju uvođenje novih raspodijeljenih sustava za pohranu i aplikacija. Potrebno je optimizirati pohranu različitih tipova podataka, osigurati skalabilnost čvorova za pohranu, vršiti deduplikaciju podataka za poboljšanje performansi čitanja, propusnosti i učinkovitosti. Problemi pohrane velike količine strukturiranih i nestrukturiranih podataka uključuju i integraciju, pohranu i analizu nestrukturiranih podataka iz različitih raspodijeljenih izvora podataka.

Tehnologije velikih količina podataka moraju odgovoriti i na ključna pitanja sigurnosti, upravljanja i privatnosti što ima utjecaj na obradu i pohranu velikih količina povjerljivih podataka. Uz pohranu, izazovi velikih količina podataka su i [73]:

- **obrada podataka** - potrebni su odgovarajući računalni resursi (CPU, mreža, pohrana) i učinkoviti skalabilni algoritmi za redukciju podataka, uklanjanje irelevantnih, neispravnih podataka i osiguravanje pouzdane obrade osjetljivih podataka te njihovu primjenu u stvarnom vremenu
- **upravljanje podacima** - optimiziran pristup podacima, učinkovita raspodjela sadržaja, vizualizacija, analiza, integracija i upravljanje velikim količinama nestrukturiranih heterogenih podataka korištenjem novih tehnologija za organizaciju podataka
- **prijenos podataka** - različite faze obrade podataka i njihov prijenos od senzora do mjesta za pohranu, integriranje iz višestrukih podatkovnih centara, prijenos do platformi za obradu i prijenos do čvorova za analizu što zahtijeva prethodnu obradu podataka algoritmima za redukciju veličine podataka prije prijenosa
- **latencija i mrežna propusnost** s obzirom na lokaciju podataka
- **sigurnost i privatnost** - fundamentalno pitanje za osiguravanje pouzdanosti, integriteta pristupa nestrukturiranim i heterogenim podacima koji traže nove enkripcijske standarde, algoritme, metodologije i sustave upravljanja podacima koji će uspješno odgovoriti na skalabilnost i tražene performanse i osigurati privatnost korisnika (zdravstveni podaci, društvene mreže i sl.)
- **kvaliteta podataka** - točnost, potpunost, redundancija i dosljednost promjenjivih podataka koje koristi više sudionika
- **standardizacija** primjenjivih rješenja i tehnologija
- **redukcija potrošene energije** u podatkovnim centrima.

Postojeće tehnike i alati velikih količina podataka ograničeni su i nisu u mogućnosti u potpunosti riješiti različite probleme ovog područja koje ima golem potencijal. Potrebna su sklopovska i programska rješenja i naprednije tehnike pohrane, zahtjevne obrade podataka i I/O i naprednije računalne arhitekture. Za donošenje odluka na temelju podataka, otkrivanje i predviđanje znanja koristi se umjetna inteligencija, algoritmi strojnog učenja i rudarenja podataka.

Razvoj Hadoop i Spark platformi pokrenuo je novu industriju aplikacija i usluga. Takva rješenja za pohranu podataka koriste podatkovne i sustave datoteka i skladišta objekata u oblaku kao što su HDFS, Ceph, Open Stack Swift ili Amazon S3.

Pregledom znanstvenih radova u području optimizacije pohrane podataka analizirano je nekoliko pristupa.

Autori u [74] predlažu korištenje hijerarhijski proširenog mehanizama pohrane za obradu velikih dinamičkih podataka, uz pohranu podataka prema tipu podataka i dinamičko dodavanje čvorova za pohranu.

Tan i dr. [75] korištenjem deduplikacije podataka postižu delinearizaciju postavljanja podataka, što ponekad utječe na performanse čitanja, propusnost i učinkovitost. Predloženo je smanjenje omjera kompresije za učinkovito smanjivanje linearizacije rasporeda podataka. Deduplikacija enkriptiranih podataka značajno utječe na ostvarivanje i performanse pohrane u oblaku [76].

U [77] autori predlažu nekoliko pristupa za poboljšanje učinkovitosti pohrane i pristupa malim datotekama za pohranu u oblaku, uključujući predloške grupiranja datoteka ili spajanja datoteka i predupozorenja.

BlobSeer [78] je skalabilan raspodijeljeni sustav pohrane koji optimizaciju provodi decentralizacijom metapodataka i kontrolama konkurentnosti zasnovanim na verzijama koje omogućuju pisanje s proizvoljnim odstupanjima, postižući paralelizam na razini podataka.

Raspodjela metapodataka na nekoliko poslužitelja ublažava usko grlo centraliziranog upravljanja za sustave koji uključuju pristup većem broju malih datoteka, kada latencija za pristup metapodacima može biti važna u ukupnom vremenu pristupa podacima. Dvije glavne metode koje se koriste za raspodjelu opterećenja među poslužiteljima metapodataka u postojećim raspodijeljenim datotečnim sustavima su particioniranje i raspršivanje (engl. *hashing*). S obzirom da diskovni sustavi za on-line pohranu nisu dovoljno skalabilni, fokus se prebacuje na RAM memoriju koja koristi disk za arhiviranje sigurnosnih kopija (BigTable, Cassandra). Sustavi memorijskih baza podataka važno su rješenje za podršku upravljanja transakcijama na *streaming* aplikacijama. Većina prikupljenih podataka treba biti obrađena brzo, na *online (streaming)* način što nameće potrebu za postizanjem niske latencije i visoke propusnosti [43].

Također, kognitivna pohrana [79] može poboljšati efikasnost sustava za pohranu stavljajući značaj na vrijednost i važnost podataka za dizajn sustava. Nužno je implementirati alate i metode za analizu koji bi kontinuirano procjenjivali vrijednost i značaj podataka u određenom trenutku i predviđali varijacije konteksta za prilagođavanje metrika o kojima ovisi proces klasifikacije podataka, a i njihova sigurnost. Autori u [88] su koristili algoritme strojnog učenja i pohranu u oblaku za poboljšanje i ubrzanje klasifikacije objekata iz astronomskog seta podataka.

Jedna od aktualnih tema je i optimizacija obrade povezanih podataka strukture grafa [89], što će zahtijevati i nove modele njihove pohrane.

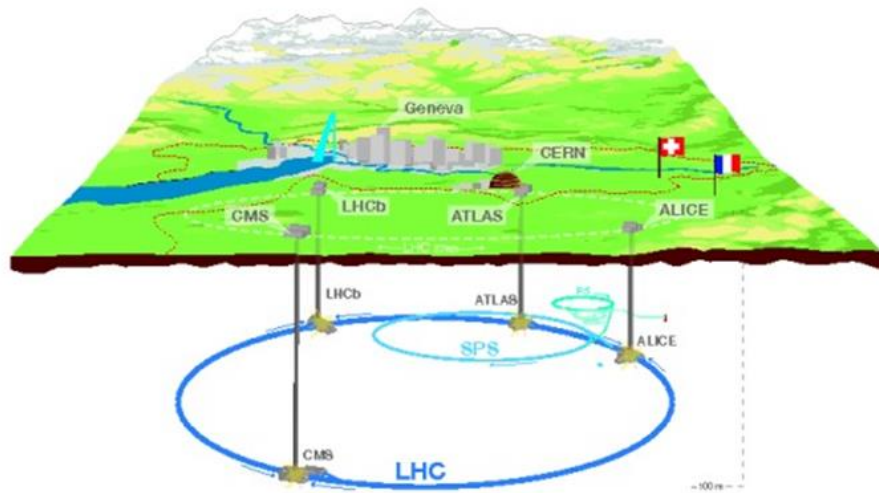
Mnogi sustavi u cjelokupno rješenje integriraju HDFS koji je gotovo postao standard za skalabilnu i pouzdanu pohranu velikih količina podataka. Ograničenja HDFS-a su korištenje metapodataka na imenskom čvoru, nemogućnost nasumičnog pisanja, smanjena učinkovitost pohrane i pristupa malim datotekama.

Postoji veliki prostor za poboljšanja u sloju za pohranu kako bi se bolje zadovoljile potrebe današnjih aplikacija. Cilj je omogućiti obradu u stvarnom vremenu blizu mjesta na kojem se generiraju podatkovni tokovi (računalstvo u magli), dok bi se oblak koristio za daljnju, vremenski nekritičnu obradu i arhiviranje podataka. Računalstvo u oblaku kao visoko skalabilna platforma raspodijeljenog računalstva ima potencijal za pružanje resursa za razvoj i osiguravanje sigurne pohrane, obrade i analizu heterogenih podataka uz uravnoteženo opterećenje što predstavlja veliki izazov. Računalstvo u oblaku i područje velikih količina podataka razvijaju se paralelno kako bi omogućili multidisciplinarnu znanost i društvo. Pojavljuju se novi modeli usluga u oblaku kao baza podataka kao usluga (engl. *Database as a Service*) ili velike količine podataka kao usluga (engl. *Big Data as a Service*).

7.1. CERN – najveći svjetski znanstveni laboratorij

Primjer izvora velike količine podataka je najveći svjetski znanstveni laboratorij CERN koji ima četiri velika LHC [80] (hrv. Veliki hadronski sudarivač) eksperimenata, ATLAS, ALICE, CMS i LHCb, (Slika 7.1.) koji svakodnevno prikupljaju i generiraju velike količine podataka pomoću visoko propusnih senzora i instrumenata. U jednoj sekundi stvori se i do 1 PB podataka iz nekoliko milijardi sudara čestica koje se kreću približno brzinom svjetlosti. Nakon redukcije podataka, potrebno je godišnje pohraniti i obraditi oko 50 PB podataka. Dnevno se obavlja oko 2000000 zadataka, a prijenos podataka se odvija brzinama oko 35GB/s.

Trenutno, CERN-ov podatkovni centar ima 15000 poslužitelja, 230000 procesorskih jezgri, više od 90000 diskova za pohranu 280 PB podataka i više od 30000 vrpci za dugotrajnu pohranu 400 PB podataka [81]. Zahtjevna pohrana i obrada takvih podataka temelji se na spletu računala kojeg uz navedene resurse (20% ukupnih resursa) čini i više od 170 velikih podatkovnih centara razmještenih u 42 države diljem svijeta s kojima zajedno tvori WLCG (*Worldwide LHC Computing Grid*) [82]. WLCG računalni splet sustav je organiziran u 4 razine (engl. *tier*) koji obavljaju specifične zadatke obrade.



Slika 7.1. LHC eksperimenti na CERN-u. Preuzeto s [85].

Prikupljeni podaci se kao datoteke u ROOT [83] formatu pohranjuju u EOS sustav, visoko skalabilan raspodijeljeni datotečni sustav za diskovnu pohranu niske latencije. Uz EOS, za dugotrajnu pohranu LHC eksperimenti koriste i CASTOR sustav vrpce [84] (Slika 7.2.). Većina podataka, izvorni (engl. *raw*) podaci, logovi poslužitelja, izvorni logovi uređaja, je nestrukturirana. Podaci dobiveni analizom su često polustrukturirani, a podaci za izvještaje strukturirani.

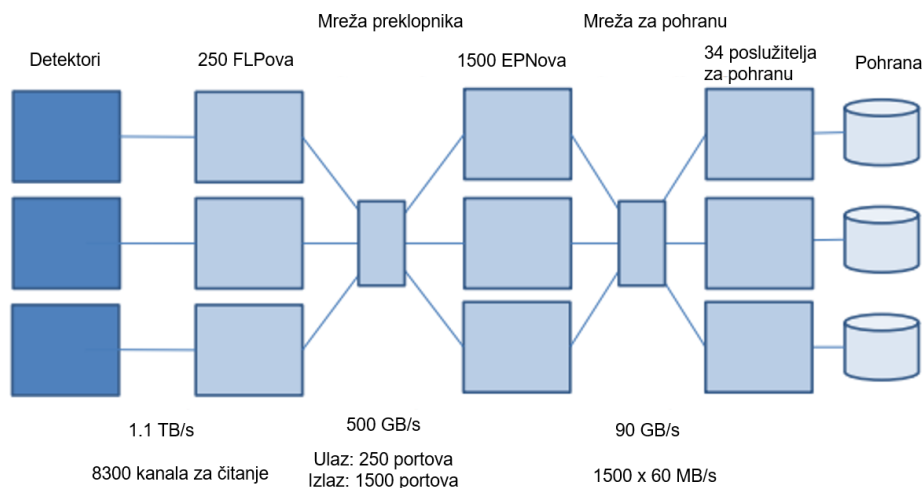
U budućnosti dostupni resursi neće biti dovoljni jer količine prikupljenih izvornih nestrukturiranih podataka rastu, a nakon provedene nadogradnje eksperimenata [86] očekuje se znatan porast brzina stvaranja i količina važnih podataka. Očekuje se da će potrebe za resursima do 2027. porasti i do sto puta, a količine proizvedenih podataka će biti reda EB. Stoga, bit će potrebno nadograditi podatkovni sustav i unaprijediti algoritme pohrane i obrade. Korištenjem novih platformi otvorenog koda potrebno je optimizirati i reducirati korištenje resursa. CERN je već počeo razvijati svoje usluge u oblaku [87].

Za istaknuti je rad doktorandice na CERN-u u okviru ALICE eksperimenta na razvoju pojednostavljenih modela za evoluciju ALICE računalnih resursa na računalnim modelima za *Run 2* i *Run 3* korištenjem Google proizvoda koji bi se izvršavao na Google Cloudu.



Slika 7.2. Pohrana podataka LHC eksperimenta. Prilagođeno prema [91].

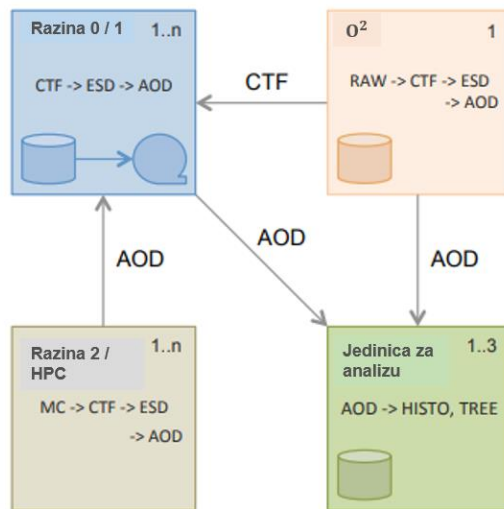
Kroz proces nadogradnje ALICE eksperimenta za razdoblje od 2019. do 2021., osim nadogradnje detektora, jedan od ciljeva je i nadogradnja računalnih kapaciteta kroz izgradnju O^2 (engl. *Online-Offline*) računalnog sustava za maksimalnu kompresiju podataka kao dodatak postojećem računalnom splet sustavu (Slika 7.4.). O^2 sustav će paralelno s prikupljanjem podataka, izvoditi *online* redukciju podatkovnog volumena kroz djelomičnu kalibraciju, rekonstrukciju podataka i izvorne podatke zamijeniti komprimiranim podacima. To zahtijeva razvoj programske podrške i korištenje akceleratora obrade (FPGA, GPU). O^2 farma će se sastojati od 250 FLP (engl. *First Level Processor*) i 1500 EPN (engl. *Event Processing Node*) računalnih čvorova (Slika 7.3.). FLP-ovi će prikupljati izvorne podatke detektora brzinom od 1.1 TB/s preko 8300 *read-out* linkova (DDL i GBT based Frontend). Svaki FLP će dobiti podatke od do 48 optičkih veza do 3.2 Gb/s. Podaci iz tih tokova bit će komprimirani, spojeni, podijeljeni u vremenske okvire (engl. *Sub-Time Frame*) i privremeno spremljeni do slanja EPN-u. EPN će prikupljati i rekonstruirati podatke svakog detektora te dodatno komprimirati podatke što će rezultirati stvaranjem vremenskih okvira za pohranu (engl. *Time Frame*). FLP-ovi će imati I/O kapacitet reda veličine 100 Gb/s kao vršni ulaz, 50 Gb/s kao stalni ulaz i 20 Gb/s kao stalni izlaz. Ukupni protok podataka do njihove pohrane doseže 90 GB/s što čini lokalnu propusnost pisanja na disk od 60MB/s po EPN-u nakon kompresije.



Slika 7.3. Sklopovski prikaz toka podataka O²sustava. Prilagođeno prema [86].

Razina 0 (engl. *Tier 0*, CERN računalni centar) postojećeg računalnog spleta vršit će rekonstrukciju i kalibraciju nad dijelom podataka te po potrebi i Monte Carlo simulaciju. Razina 1 (engl. *Tier 1*) kao i 0, pružat će CPU resurse te resurse za pohranu na diskovima i vrpcama, te izvoditi rekonstrukciju, kalibraciju i simulaciju nad vlastitim dijelom pohranjenih komprimiranih podataka. MC simulacija predviđena je za izvođenje na računalima razine 2 (engl. *Tier 2*). Analizu podataka izvodit će *Analysis Facility (AF)*. Uz pohranu prikupljenih i obrađenih podataka, potrebno je pohraniti i informacije potrebne za rekonstrukciju i simulaciju, a za to je predviđena baza podataka (engl. *Condition and Calibration Database, CCDB*). Razina 0 već se razvija prema modelu privatnog oblaka IaaS, a u planu je razvijanje i ostalih razina kao regionalnih oblaka i korištenje virtualizacije fizičkih resursa za osiguravanje resursa za pohranu i obradu.

Sklopovski, O² sustav se zasniva na diskovnoj RAID tehnologiji za potrebe osiguravanja kapaciteta pohrane od 70 PB i propusnosti za čitanje i pisanje od 90 GB/s. Programski, uzimajući u obzir performanse i redundanciju, razmatra se korištenje grozd datotečnog sustava (LustreFS i GPFS), kao i objektna pohrana (CEPH, RADOS, EOS) [86].



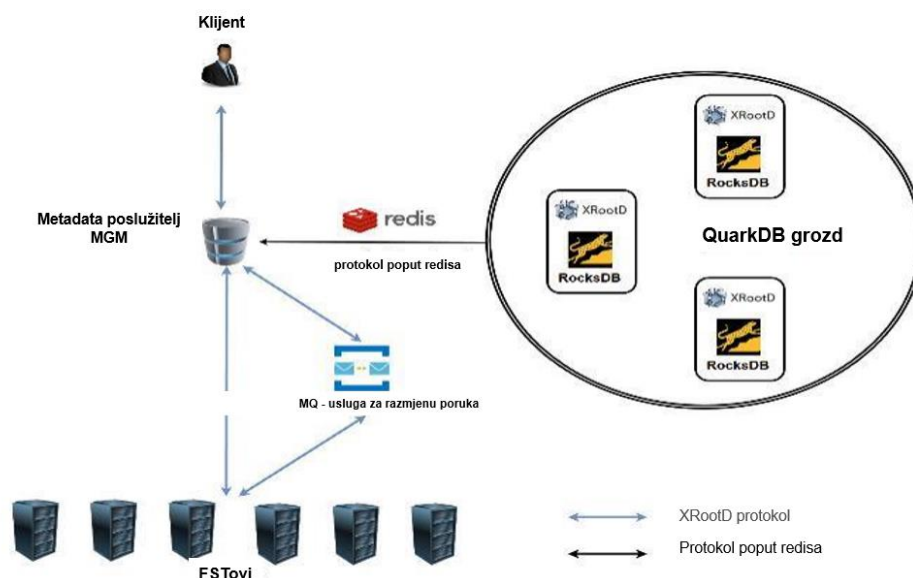
Slika 7.4. Tok podataka među sastavnicama O² sustava i razinama WLCG računalnog spleta.

Prilagođeno prema [86].

Objektna pohrana podataka udovoljava visokim zahtjevima za pohranu nestrukturiranih i nepromjenjivih podataka te se primjenjuje za pohranu web sadržaja i računalstva u oblaku omogućujući jednostavno skaliranje volumena i propusnosti. Takvi sustavi pohrane imaju ravan imenski prostor i jednostavno sučelje koje podržava dodavanje parova jedinstvenih ključeva i obnovljivih velikih binarnih objekata (BLOB) što ih razlikuje od raspodijeljenog sustava datoteka i NoSQL baza podataka. Sustav za pohranu više PB mora biti vrlo skalabilan bez bilo koje točke neuspjeha i gubitka podataka uslijed kvarova, programskih greški, operativnih pogrešaka i sl. Prednosti korištenja objektna pohrane za O² sustav su ulazni podaci veličine 6 GB te manja količina metapodataka kojima treba upravljati.

CEPH je besplatna platforma za pohranu koja omogućava objektnu, blokovsku i datotečnu pohranu unutar raspodijeljenog računalnog grozda, visoku skalabilnost, dostupnost, pouzdanost te postizanje odličnih performansi. Najisplativija implementacija CEPH modela je korištenje računalnih čvorova na kojima se uz obradu odvija i pohrana podataka na diskovima. Objektna komponenta pohrane u CEPH-u, RADOS (engl. *Reliable Autonomic Distributed Object Store*) već je implementiran u podatkovnim centrima unutar LHC-a računalnog spleta.

ALICE razvija postojeći model upravljanja podacima temeljen na mreži XRootD elementima za pohranu prema imenskom prostoru EOS sustava (Slika 7.5.).



Slika 7.5. EOS arhitektura. Prilagođeno prema [92].

EOS je implementiran u C++ koristeći XRootD okvir koji omogućuje pristup i prijenos podataka po principu klijent/poslužitelj. EOS čine: čvorovi za pohranu datoteka (engl. *File Storage Node*, FST) koji upravljaju fizičkim diskovima i poslužuju datoteke, poslužitelji metapodataka (engl. *Metadata Server*, MGM) koji upravljaju imenskim prostorom i preusmjeravaju klijente na FST te red čekanja na poruke (engl. *Message Queue*, MQ), zadužen za komunikaciju među grozdovima i promjene konfiguracije. Primjer primjene heterogenih tehnologija za pohranu je EOS koji uz osnovne čvorove, implementira i QuarkDB grozd. QuarkDB je riješio problem ograničenog kapaciteta RAM memorije i vrijeme pokretanja sustava, glavne nedostatke *memorijskog* imenskog prostora CERN-ov QuarkDB pokreće se kao dodatak XRootD [93] poslužiteljskom okviru kojeg koristi EOS, a razvijen je na osnovi NoSQL baze RocksDB koja je temeljni spremnik za pohranu. S EOS-om komunicira putem Redis protokola. Također, EOS za pohranu koristi i usluga u oblaku dostupna svim CERN-ovim korisnicima, CERNBox [95], zasnovana na platformi otvorenog koda ownCloud [94].

Očekuje se da će potrebe za pohranom podataka biti reda EB - red veličine više nego danas. To stvara niz problema u upravljanju podacima, analizi i obradi koje i Google smatra izazovnim. Izazovima u vezi s podacima LHC-a eksperimenata za razdoblja 2021.-2023. i 2026.-2029. trebat će se pristupati ne samo uslugama računalstva u oblaku. Tehnologije kvantnog računalstva obećavaju znatno ubrzanje računski zahtjevnih zadataka. Pouzdanost diskova i integritet podataka je važno pitanje jer obnova volumena nakon kvara na disku može dugo potrajati. Potrebni su viši pariteti, stoga klasični RAID sustavi više nisu opcija i zamijenit će ih drugi sustavi koji osiguravaju više razine pariteta.

Resursi računalstva u oblaku mogu se koristiti na prilagodljiv i hibridan način korištenjem novih načina i tehnologija velikih podataka u području fizike visokih energija (HEP). Tehnologije pohrane morat će se uskladiti s odgovarajućim računalnim resursima koje eksperimenti zahtijevaju te ostati dovoljno fleksibilni da podrže evoluirajuće fizičke mogućnosti pohrane. Potrebno je ispitati standarde obrade za opću namjenu za HEP obradu podataka i otvoriti HEP područje zajednici znanstvenika u području podatkovne znanosti. Prostora za istraživanje ima u redukciji količine podataka koja se stvara dnevno i sortiranju kroz različite pristupe formatiranja, pisanja i pristupa podacima korištenjem novih tehnologija kao što je Spark (npr. pri analizi EOS logova o transakcijama i izvršenim operacijama na EOS sustavu ili HL-LHC analizi podataka u ROOT datotekama pohranjenim u EOS-u na HPC sustavima). Za postizanje maksimalnih performansi na novim sklopovskim platformama potrebno je modernizirati kod. Strojno učenje ima potencijalno veliku ulogu u praćenju akceleratorskog lanca i optimiziranju korištenja resursa.

8. ZAKLJUČAK

Podaci su temelj konkurentnosti u digitalnom razdoblju. Velike količine podataka imaju najvažniju ulogu u sustavima pohrane za podatkovno intenzivno računalstvo. Podaci su namijenjeni pohrani, dugotrajnom čuvanju i prosljeđivanju kreiranih i prikupljenih podataka kako bi se njihovom analizom došlo do izvornih znanstvenih rezultata. Velike količine podataka treba urediti prema standardima određenog područja istraživanja kako bi skup bio upotrebljiv. Visoka učinkovitost i ekonomična rješenja za pohranu važna su u području povećanja podataka i usluga upravljanja podacima.

Razvoj usluga i korištenje novih računalnih platformi omogućit će konkurentnija istraživanja, veće kapacitete skladištenja, intenzivnu obradu podataka i smanjenje cijena usluga.

Analizom dosadašnjih objavljenih znanstvenih istraživanja utvrđeno je da je prilagodba i razvoj modela pohrane rastućih količina heterogenih podataka u fokusu istraživanja mnogih znanstvenika. Daljnja istraživanja trebalo bi temeljiti na istraživanju metoda za poboljšanje pohrane rastućih količina podataka u oblaku.

U ovom radu dat je pregled koncepata i tehnologija za pohranu velikih količina podataka. Osnovni koncepti pohrane zasnivaju se na datotečnoj, blok i objektnoj pohrani podataka, a tehnologije se mogu podijeliti na diskovne (RDBMS, NoSQL, NewSQL i DFS) ili memorijske (IMDG i IMDB). Opisane su i značajke podataka (strukturirani, nestrukturirani, polustrukturirani i metapodaci) koji nastaju iz raznih izvora. Kronološki su predstavljene podatkovno intenzivne paradigme (grozd računala, splet računala, računalni oblak, računalstvo u magli i rosno računalstvo). Obrađena su i područja obrade i analize velikih količina podataka. Temeljem pregleda znanstvenih radova, razmatrani su problemi, mogućnosti i izazovi velikih količina podataka te definirana i analizirana područja istraživanja koja još uvijek nude pregršt izazova i neriješenih problema.

Naveden je najveći svjetski znanstveni istraživački centar CERN kao primjer izvora velike količine podataka koji kroz LHC eksperimente svakodnevno proizvede nekoliko PB podataka. Navedeni su i procesi i ciljevi nadogradnje za pohranu velikih količina podataka velikog CERN-ovog ALICE eksperimenta.

Dakle, u ovom radu sintetizirano je stanje područja pohrane velikih količina podataka koje je trenutno u vrlo dinamičnim procesima. Detektirani su problemi, izneseni zaključci i predstavljena platforma s naznakama i smjericama za budući rad.

U pisanoj prezentaciji rada nastojalo se ispravno koristiti strukovno nazivlje uz prikladno proširenje strukovnog nazivlja iz odabranog područja u duhu hrvatskog standardnog jezika. Korištene preuzete slike s izvornog prevedene su na hrvatski jezik.

U narednoj fazi doktorskog studija planiraju se provesti potrebna istraživanja s ciljem pronalaženja pragmatičnih rješenja s rezultatima u sferi pohrane velikih količina podataka, s determiniranom okosnicom, odnosno predmetom istraživačkog procesa.

LITERATURA

- [1] Moore, G.: „Cramming More Components onto Integrated Circuits”, *Electronics*, Vol. 38, No. 8, pp. 114-117, 1965.
- [2] Sterling, T. L. i dr.: „Beowulf: A Parallel Workstation for Scientific Computation”, *Proceedings of 24th International Conference on Parallel Processing*, pp. 11-14, 1995.
- [3] Torelli, J.C. i dr.: „A high performance 3D exact Euclidean distance transform algorithm for distributed computing”, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 24, No. 6, pp. 897-915, 2010.
- [4] Kornobis, E. i dr.: „TRUFA: A User-Friendly Web Server for de novo RNA-seq Analysis Using Cluster Computing”, *Evolutionary Bioinformatics Online*, Vol. 11, pp. 97-104, 2015.
- [5] Foster, I. i Kesselman C.: „The Grid: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann Publishers, 1st edition, ISBN-10: 1558604758, San Francisco, USA, 1998.
- [6] S Interneta, <http://dis-sys.blogspot.com/2011/07/grid-computing-system.html>, pristup: 27. siječnja 2019.
- [7] Simmon, E.: „Evaluation of Cloud Computing Services Based on NIST SP 800-145”, National Institute of Standards and Technology, Information Technology Laboratory, 2018., <https://doi.org/10.6028/NIST.SP.500-322>, pristup: 09. svibnja 2018.
- [8] Gastermann, B. i dr.: „Secure Implementation of an On-Premises Cloud Storage Service for Small and Medium-Sized Enterprises”, *Procedia Engineering*, 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, pp. 574-583, 2014.
- [9] D’Agostino, D. i dr.: „Cloud Infrastructures for In Silico Drug Discovery: Economic and Practical Aspects”, *BioMed Research International*, Vol. 2013, Article ID 138012, 19 pp., 2013.
- [10] Kagadis, G. C. i dr.: „Cloud computing in medical imaging, *Medical Physics*”, Vol. 40, No. 7, 2013.
- [11] Morabito, R.; Kjallman, J. i Komu, M.: „Hypervisors vs. Lightweight Virtualization: A Performance Comparison”, *Proceedings of the IEEE International Conference on Cloud Engineering*, pp. 386–393, 2015.
- [12] S Interneta, <https://blogs.technet.microsoft.com/yungchou/2010/11/15/cloud-computing-primer-for-it-pros/>, pristup: 27. siječnja 2019.
- [13] Duan Y. C. i dr.: „Exploring the Categories and Models of Everything as a Service (XaaS)”, *International Journal of Grid and Distributed Computing*, Vol. 8, No. 6, pp. 215 - 228, 2015.

- [14] Bonomi, F. i dr.: „Fog Computing and Its Role in the Internet of Things”, Proceedings of the first edition of the Mobile Cloud Computing workshop, pp. 13–16, ACM, 2012.
- [15] Cisco: „Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are”, 2015., https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf, pristup: 25. travnja 2018.
- [16] Vilalta, R. I dr.: „TelcoFog: A unified flexible fog and cloud computing architecture for 5G networks”, IEEE Communications Magazine, Vol. 55, No. 8, pp. 36–43, 2017.
- [17] Zhu, J. i dr.: „Improving Web Sites Performance Using Edge Servers in Fog Computing Architecture”, 2013 IEEE Seventh international Symposium on Service-Oriented System Engineering, pp. 320–323, IEEE, 2013.
- [18] OpenFog Consortium Architecture Working Group: „OpenFog Reference Architecture for Fog Computing”, 2017., https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf
- [19] OpenFog Consortium, <https://www.openfogconsortium.org/>, pristup: 19. svibnja 2018.
- [20] Jalali, F. i dr.: „Fog Computing May Help to Save Energy in Cloud Computing”, IEEE Journal of Selected Areas in Communications, Vol. 34, No. 5, pp. 1728 - 1739, 2016.
- [21] Wang, Y.: „Definition and Categorization of Dew Computing”, Open Journal of Cloud Computing, Vol. 3, No. 1, pp. 1-7, 2016.
- [22] Wang, Y.: „Cloud-dew architecture”, International Journal of Cloud Computing, Vol. 4, No. 3, pp. 199-210, 2015.
- [23] Skala, K.: „Scalable Distributed Computing Hierarchy: Cloud, Fog and Dew Computing”, Open Journal of Cloud Computing, Vol. 2, No. 1, pp. 16-24, 2015.
- [24] Digital Economy Forum, „How Your Digital Footprint Generates Big Data”, <https://digitaleconomyforum.org/chapter-2-how-your-digital-footprint-generates-big-data/>, pristup: 27. siječnja 2019.
- [25] S Interneta, <https://i2.wp.com/caringo.com/wp-content/uploads/2015/06/Object-File-Block-Storage.png>, pristup: 27. siječnja 2019.
- [26] S Interneta, <https://www.mysql.com/>, pristup: 27. siječnja 2019.
- [27] S Interneta, <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>, pristup: 27. siječnja 2019.
- [28] Brewer, E.: „CAP twelve years later: How the “rules” have changed. Computer”, Vol. 45, No. 2, pp. 23 - 29, 2012.
- [29] Oracle NoSQL Database, <https://www.oracle.com/technetwork/database/database-technologies/nosqldb/overview/index.html>, pristup: 28. siječnja 2019.

- [30] Dynamo Paper, <https://www.dynamodbguide.com/the-dynamo-paper/>, pristup: 28. siječnja 2019.
- [31] Riak, <http://basho.com/products/>, pristup: 28. siječnja 2019.
- [32] Cassandra, <http://cassandra.apache.org/>, pristup: 28. siječnja 2019.
- [33] HBase, <https://hbase.apache.org/>, pristup: 28. siječnja 2019.
- [34] Chang, F. i dr. „Bigtable: A Distributed Storage System for Structured Data”, 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 205-218, 2016.
- [35] Neo4J, <https://neo4j.com/>, pristup: 28. siječnja 2019.
- [36] InfiniteGraph, <https://www.objectivity.com/products/infinitegraph/>, pristup: 28. siječnja 2019.
- [37] OrientDB, <https://orientdb.com/>, pristup: 28. siječnja 2019.
- [38] MongoDB, <https://www.mongodb.com/>, pristup: 28. siječnja 2019.
- [39] Terrastore, <https://code.google.com/archive/p/terrastore/>, pristup: 29. siječnja 2019.
- [40] CouchDB, <http://couchdb.apache.org/>, pristup: 28. siječnja 2019.
- [41] XQuery, https://www.w3schools.com/xml/xquery_intro.asp, pristup: 29. siječnja 2019.
- [42] Harris, S. i Seaborne, A.: „SPARQL 1.1 Query Language”, 2013.,
<http://www.w3.org/TR/2013/REC-sparql11-query-20130321/>, pristup: 29. siječnja 2019.
- [43] Grolinger, K. i dr.: „Data management in cloud environments: NoSQL and NewSQL data stores”, Journal of Cloud Computing: Advances, Systems and Applications, 2013, 2:22
- [44] Corbett, J. C. i dr.: „Spanner: Google’s Globally-Distributed Database”, In Tenth USENIX Symposium on Operating Systems Design and Implementation (OSDI 12), pp. 261–264, 2012.
- [45] NuoDB, <https://www.nuodb.com/>, pristup: 29. siječnja 2019.
- [46] Apache Ignite, <https://ignite.apache.org/>, pristup: 29. siječnja 2019.
- [47] Hadoop HDFS, https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html, pristup: 29. siječnja 2019.
- [48] Apache Hadoop, <http://hadoop.apache.org/docs/stable/>, pristup: 29. siječnja 2019.
- [49] CephFS, <http://docs.ceph.com/docs/master/architecture/#arch-cephfs>, pristup: 29. siječnja 2019.
- [50] MooseFS, <https://moosefs.com/>, pristup: 29. siječnja 2019.
- [51] GlusterFS, <https://docs.gluster.org/en/latest/>, pristup: 29. siječnja 2019.
- [52] Ruzzi, J.: „Oracle Coherence Developer's Guide, Release 3.6.1”, 2010.,
https://docs.oracle.com/cd/E15357_01/coh.360/e15723/toc.htm, pristup: 29. siječnja 2019.
- [53] Oracle Coherence,
<https://www.oracle.com/technetwork/middleware/coherence/overview/index.html>, pristup: 29. siječnja 2019.

- [54] Infinispan, <http://infinispan.org/>, pristup: 29. siječnja 2019.
- [55] Hazelcast, <https://hazelcast.com/products/imdg/>, pristup: 29. siječnja 2019.
- [56] Exasol, <https://www.exasol.com/en/>, pristup: 29. siječnja 2019.
- [57] ArangoDB, <https://www.arangodb.com/>, pristup: 29. siječnja 2019.
- [58] Redis, <https://redis.io/>, pristup: 29. siječnja 2019.
- [59] Apache Ignite, <https://ignite.apache.org/>, pristup: 29. siječnja 2019.
- [60] VoltDB, <https://www.voltdb.com/>, pristup: 29. siječnja 2019.
- [61] Dean, J. i Ghemawat, S.: „MapReduce: Simplified Data Processing on Large Clusters”, OSDI '04: 6th Symposium on Operating Systems Design and Implementation, 2004.
- [62] Apache Spark, <https://spark.apache.org/>, pristup: 29. siječnja 2019.
- [63] Apache YARN, <http://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html>, pristup: 29. siječnja 2019.
- [64] Apache Spark, „Cluster Mode Overview“, <https://spark.apache.org/docs/latest/cluster-overview.html>, pristup: 29. siječnja 2019.
- [65] Storey, V. C. i Song, I.: „Big data technologies and management: What conceptual modeling can do”, *Data&Knowledge Engineering*, Vol. 108, pp. 50 - 67, 2017.
- [66] Schnase, J. L. i dr.: „Big Data Challenges in Climate Science: Improving the next-generation cyberinfrastructure”, *IEEE Geoscience and Remote Sensing Magazine*, Vol. 4, No.3, pp. 10-22, 2016.
- [67] Gutsche, O. i dr.: „Big Data in HEP: A comprehensive use case study”, *Journal of Physics: Conference Series*, 898, 072012, 2017.
- [68] Chen, H., Chiang, R. H. L. i Storey, V. C.: „Business Intelligence and Analytics: From Big Data to Big Impact”, *MIS quarterly*, Vol. 36, No. 4, pp. 1165-1188, 2012.
- [69] Yang, C. T. i dr.: „Implementation of Big Data Accessing and Processing Platform for Medical Records in Cloud”, *Journal of Medical Systems*, Vol. 41:149, pp. 1-28, 2017.
- [70] Chen, J. i dr.: „A Disease Diagnosis and Treatment Recommendation System Based on Big Data Mining and Cloud Computing”, *Information Sciences*, Vol. 435, pp. 124-149, 2018.
- [71] Rodrigues, A. P. i Chiplunkar, N. N.: „Real-time Twitter data analysis using Hadoop ecosystem”, *Cogent Engineering*, Vol. 5, No. 1, pp. 1-16, 2018.
- [72] Kruger, N. i Teuteberg, F.: „From Smart Meters to Smart Products: Reviewing Big Data driven Product Innovation in the European Electricity Retail Market”, Cunningham, D. W., Hofstedt, P., Meer, K. & Schmitt, I. (Hrsg.), *INFORMATIK 2015. Bonn*, pp. 1171-1182, 2015.
- [73] Yang, C. i dr.: „Big Data and cloud computing: innovation opportunities and challenges”, *International Journal of Digital Earth*, Vol. 10, No. 1, pp. 13-53, 2017.

- [74] Wang, Y. i dr.: „Using Elasticity to Improve Inline Data Deduplication Storage Systems”, CLOUD '14 Proceedings of the 2014 IEEE International Conference on Cloud Computing”, pp. 785-792, 2014.
- [75] Tan, Y. i dr.: „Reducing the De-linearization of Data Placement to Improve Deduplication Performance”, SSC '12 Proceedings of the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, pp. 796 -800, 2012.
- [76] Yan, Z. i dr.: „Deduplication on Encrypted Big Data in Cloud”, IEEE Transactions on Big Data, Vol. 2, No. 2, pp. 136-150, 2016.
- [77] Dong, B. i dr.: „An optimized approach for storing and accessing small files on cloud storage”, Journal of Network and Computer Applications, Vol. 35, pp. 1847-1862, 2012.
- [78] Nicolae, B. i dr.: „BlobSeer: Next-generation data management for large scale infrastructures”, Journal of Parallel and Distributed Computing, Vol. 71, No. 2, pp. 169-184, 2011.
- [79] Cherubini, G. i dr.: „Cognitive Storage for Big Data”, IEEE Computer, Vol. 49, No. 4, pp. 43-51, 2016.
- [80] S Interneta, <https://home.cern/science/experiments>, pristup 1. veljače 2019.
- [81] S Interneta, <http://monit-grafana-open.cern.ch/d/000000884/it-overview?orgId=16>, pristup: 1. veljače 2019.
- [82] S Interneta, <http://wlcg.web.cern.ch/>, Worldwide LHC Computing Grid, pristup: 1. veljače 2019.
- [83] S Interneta, <https://root.cern.ch/>, ROOT, pristup: 1. veljače 2019.
- [84] Espinal, X. i dr.: „CERN data services for LHC computing”, Journal of Physics: Conference Series Vol. 898, p. 062028, 2017.
- [85] S Interneta, <https://cds.cern.ch/images/OPEN-PHO-CHART-2014-006-1/file?size=medium>, pristup: 1. veljače 2019.
- [86] ALICE Technical Design Report for the Upgrade of the Online-Offline Computing System, <https://cds.cern.ch/record/2011297/files/ALICE-TDR-019.pdf>, pristup: 1. veljače 2019.
- [87] Cordeiro, C. i dr.: „CERN Computing in Commercial Clouds”, Journal of Physics: Conference Series, Vol. 898, p. 082030, 2017.
- [88] Acharya, V. i dr.: „Classification of SDSS photometric data using machine learning on a cloud“, Current science, Vol. 115, No. 2, pp. 249-257, 2018.
- [89] Kalavri, V. i dr.: „High-Level Programming Abstractions for Distributed Graph Processing“, IEEE Transactions on Knowledge and Data Engineering, Vol. 30, No. 2, pp. 305-324, 2018.
- [90] Loncar, P.: “Data-Intensive Computing Paradigms for Big Data”, Proceedings of the 29th International DAAAM Symposium "Intelligent Manufacturing & Automation", pp. 1010-1018, 2018.

- [91] S Interneta, <http://eos-docs.web.cern.ch/eos-docs/intro.html>, pristup: 25. veljače 2019.
- [92] 3rd EOS workshop, <https://indico.cern.ch/event/775181/>, CERN, 2019.
- [93] XRootD, <http://xrootd.org/>
- [94] ownCloud, <https://owncloud.org/>
- [95] Mascetti, L. i dr.: „CERNBox + EOS: end-user storage for science”, Journal of Physics: Conference Series, Vol. 664, No. 6, 2015.

POPIS OZNAKA I KRATICA

ACID	Atomicity, Consistency, Isolation, Durability
ALICE	A Large Ion Collider Experiment
AOD	Analysis Object Data
API	Application Programming Interface
BASE	Basically available, soft state, eventually consistent
BLOB	Binary Large Object
CAP	Consistency, Availability and Partition Tolerance
CASTOR	CERN Advanced STORage manager
CCDB	Condition and Calibration Data Base
CERN	European Organisation for Nuclear Research
CMS	Compact Muon Solenoid
CPU	Central Processing Unit
CRUD	Create, Read, Update and Delete
CTF	Compressed Time Frame
DBiD	Database in Dew
DC	Dew-Cloud
DFS	Distributed File System
DiD	Data in Dew
EB	Exabyte
EDI	Electronic Data Interchange
engl.	engleski
EPN	Event Processing Node
ESD	Event Summary Data
FCoE	Fibre Channel over Ethernet
FLP	First Level Processor
FPGA	Field Programmable Gate Array
FST	File Storage Node
GPS	Global Positioning System
GPU	Graphics Processing Unit

HAC	High Availability Computing
HDD	Hard disk drive
HDFS	Hadoop Distributed File System
HEP	High Energy Physics
HPC	High Performance Computing
HTC	High Throughput Computing
I/O	Input/Output
IaaS	Infrastructure as a Service
IaD	Infrastructure as Dew
IMDB	In-memory Database
IMDG	In-memory Datagrid
IoT	Internet of Things
iSCSI	Internet Small Computer System Interface
IT	Information Technology
JSON	JavaScript Object Notation
LHC	Large Hadron Collider
LHCb	Large Hadron Collider beauty
M2M	Machine to Machine
MC	Monte Carlo
MGM	Metadata Servers
NAND	Not AND
NAS	Network Attached Storage
NFS	Network File System
NIST	National Institute of Standards and Technology
NoSQL	Not only SQL
O2	Online-Offline
OOB	Out-of-band
PaaS	Platform as a Service
PB	Petabyte
PiD	Platform in Dew
QoS	Quality of Service

RADOS	Reliable Autonomic Distributed Object Store
RAID	Redundant Array of Independent Disks
RAIN	Redundant Array of Independent Nodes
RAM	Random Access Memory
RDBMS	Relational database management system
RDD	Resilient Distributed Dataset
REST	Representational State Transfer
RFID	Radio-frequency identification
SaaS	Software as a Service
SCALE	Security, Cognition, Agility, Latency, Efficiency
SiD	Software in Dew
SQL	Structured Query Language
SSD	Solid-State Drive
STF	Sub-Time Frame
STiD	Storage in Dew
TF	Time Frame
TSN	Time Sensitive Networking
URL	Uniform Resource Locator
VM	Virtual Machine
WiD	Web in Dew
WLCG	Worldwide LHC Computing Grid
XaaS	Everything as a Service
XML	Extensible Markup Language
YARN	Yet Another resource Negotiator