Pregled in analiza tehnoloških skladov za implementacijo sodobnih IT-arhitektur velepodatkov

Martina Šestak, Muhamed Turkanović
Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
martina.sestak@um.si, muhamed.turkanovic@um.si

Izvleček

Dandanes se pri implementaciji sodobnih IT-arhitektur velepodatkov podjetja odločajo za uporabo različnih tehnoloških skladov, ki so bodisi odprtokodni bodisi takšni, ki jih na trgu ponujajo oblačni ponudniki, kot so Google, Microsoft, Amazon idr. Na izbiro določenega sklada vpliva nekaj različnih dejavnikov, pogosto pa se najpomembnejši izkažejo človeški viri in strošek uporabe posameznega sklada. Kot alternativa se za zmanjšanje stroškov lahko uporabijo odprtokodne tehnologije, kot je sklad Apache, vendar tudi to prinaša določene implikacije in kompromise. Sodobne arhitekture velepodatkov pa včasih vključujejo ločeni ravni za shrambo in analitiko, pri čemer se na vsaki ravni uporablja različna tehnološka rešitev (tudi znotraj posamezne ravni), a vse z namenom shranjevanja različno strukturiranih oz. nestrukturiranih podatkov in učinkovite analize le-teh. V članku predstavljamo primer dvostopenjske IT-arhitekture, optimizirane za hrambo in analizo velepodatkov. Prav tako prikazujemo orodja in rešitve znotraj izbranih treh tehnoloških skladov (Google, Amazon, Apache), s katerimi se lahko implementira omenjena arhitektura. Analiziramo lastnosti posameznih skladov ter podajamo povzetek prednosti in izzivov pri uporabi določenega sklada.

Ključne besede: IT-arhitektura, velepodatki, masovni podatki, podatkovno skladišče, podatkovne baze, širokostolpčne hrambe, tehnološki sklad

STATE-OF-THE-ART ANALYSIS OF TECHNOLOGY STACKS FOR THE IMPLEMENTATION OF MODERN BIG DATA ARCHITECTURES

Abstract

Every individual and company perceives the dimensions of big data a bit differently. Big data is not measured as 5 hard drives nor as 5 data barrels, neither is flow of data measured in litres. Big data is a mindset that forms the foundation for data-driven business. For mass data to be truly introduced to and used in business, one must first understand it. Only then can we expect business and technology to co-exist and deliver added value. In this article, I present all dimensions of big data (i.e., 5 V's) and shed light on its use in practical examples. I also present the broader ecosystem of big data and the foundations for building an environment for big data.

Keywords: IT architecture, big data, data warehouse, database, wide-column databases, technology stack

1 UVOD

Pojem velepodatkov ali masovnih podatkov, ki se že nekaj časa uporablja kot ena ključnih besed v domeni podatkovnih tehnologij, je v zadnjem desetletju pospešil razvoj novih tehnoloških rešitev za učinkovito upravljanje naraščajočih količin podatkov, ki naslavljajo izzive, kot so razširljivost, količina, hitrost, različnost in drugo. Po trenutnih statistikah se vsak dan proizvede približno 2,5 kvintiljona (1018) bajtov podatkov, sama industrija velepodatkov pa zadnjih nekaj let vedno bolj narašča in je trenutno vredna rekordnih 274 milijard ameriških dolarjev [10]. Slednje

številke potrjujejo, da morajo podjetja neprekinjeno vlagati v sodobne tehnološke rešitve, ki jim lahko olajšajo spopadanje z izzivi v obdobju velepodatkov.

Obenem pa se podjetja, ki v vsakdanjem poslovanju obravnavajo veliko količino podatkov, pogosto srečajo s potrebo po prilagoditvi obstoječih informacijskih sistemov in integracijo novih rešitev z le-temi. Pri tem se večina finančnih sredstev uporablja za namen transformacije obstoječih informacijskih rešitev ob upoštevanju sodobnih pristopov v načrtovanju IT-arhitektur IKT-sistemov. Sodobne zahteve namreč določajo, da podatke generirajo različni elementi IKT-rešitve oz. sistema, kar pomeni, da v takšnih sistemih hitro nastopi beleženje velikih količin podatkov, da se ti generirajo z veliko hitrostjo in s strani različnih virov ter da so podatki vedno bolj nestrukturirane oblike. Slednje so obstajale tudi prej, vendar v manjših količinah, a jih nismo znali obdelati oz. po njih poizvedovati, kar pa se korenito spreminja z vedno večjim prodorom tehnik umetne inteligence.

Kot največji izziv pri vpeljavi sodobnih tehnoloških rešitev za upravljanje velepodatkov podjetja izpostavljajo dodeljena finančna sredstva za takšne projekte ter tehnične izzive pri integraciji z obstoječo infrastrukturo podjetja [10]. Ne moremo pa tudi zanemariti kompleksnosti vedno bolj pomembnega področja podatkovnega inženirstva (angl. data engineering), ki je tesno povezano s področjem podatkovne znanosti (angl. data science). S tema področjema so povezani tudi trije profili strokovnjakov, in sicer podatkovni inženir ter podatkovni znanstvenik in podatkovni analitik. Slednja izvajata aktivnosti, ki so neposredno povezane z metodami umetne inteligence, strojnega učenja, podatkovnega rudarjenja oz. s področjem poročanja in vizualizacije. V primerjavi s tem pa je naloga podatkovnega inženirja gradnja in vzdrževanje logičnih in fizičnih podatkovnih modelov ter s tem povezanih podatkovnih cevovodov. Osredotočajo se na celovito IT-arhitekturo za upravljanje velepodatkov, kakor tudi s tem povezano preoblikovanje, migriranje in združevanje podatkov ter zagotavljanje kakovosti podatkov.

Za zagotavljanje ustreznega in učinkovitega upravljanja velepodatkov je treba oblikovati celotni podatkovni cevovod (en ali več) ter opredeliti posamezne rešitve kot elemente le-tega. Ta korak predstavlja včasih izziv celo za strokovnjake, saj je težko izbrati rešitve, ki bodo med seboj ujemajoče in bodo v celoti zajemale zahteve uporabnikov. Dodaten izziv je izbira ene rešitve med številnimi možnostmi, ki so na voljo na trgu za določen namen (npr. za zajem podatkov). Večja tehnološka podjetja so razvijala rešitve, s katerimi so želela rešiti specifične izzive velepodatkov, s katerimi se srečujejo. Takšen pristop je predstavljal veliko različnih tehnologij in orodij na trgu, pri čemer so razlike med njimi v najmanjši meri oz. jih ni možno ločiti brez podrobne analize rešitev. Posledično izbira tehnološkega sklada pri projektu implementacije arhitekture za upravljanje velepodatkov ni odvisna le od stroškov implementacije, temveč tudi od drugih dejavnikov, kot so potreba po realnočasovni obdelavi, enostavnost integracije z obstoječimi IKT-rešitvami v podjetju, način interakcije s sistemom, časovna zahtevnost vzpostavitve cevovoda itn.

V nadaljevanju bomo predstavili osnovne izzive sistemov za upravljanje velepodatkov ter način naslavljanja teh izzivov sodobne arhitekture. Poglobili se bomo v lastnosti rešitev znotraj tehnoloških skladov, ki jih v sklopu svojih oblačnih storitev ponujajo podjetja Google in Amazon ter odprtokodne alternativne rešitve znotraj sklada Apache. Podrobna analiza zmogljivosti izbranih tehnoloških skladov lahko pomaga pri izbiri določenega tehnološkega sklada za implementacijo na praktičnih primerih.

2 ZAHTEVE IN IZZIVI SISTEMOV ZA UPRAVLJANJE VELEPODATKOV

Sodobni sistemi za upravljanje velepodatkov se srečajo s številnimi izzivi pri zagotavljanju funkcionalnosti, ki jih zahteva določena domena uporabe. Ena izmed od funkcionalnosti je zagotavljanje sledljivosti vsakega podatkovnega toka [9], in sicer beleženje vsake spremembe nad posameznim podatkovnim zapisom v toku (tj. pogosto v dnevniške datoteke). S tem se pridobi večji nadzor nad podatki, ki tečejo v sistemu in omogoči transparentnost celotnega cevovoda.

Kot največji izziv v obdobju velepodatkov se vedno omenjata varnost in kakovost podatkov [3, 28], za kateri je pomembno vzpostaviti ustrezne mehanizme za upravljanje podatkov v sistemu (angl. data governance), kar npr. vključuje nadzor nad uporabo podatkov (tj. kdo, kaj, kdaj, kje in za kateri namen uporablja določeni podatkovni zapis). Varnost in zasebnost podatkov sta kot izziv izpostavljeni tudi v [1], kjer so avtorji predstavili pregled najbolj pogosto omenjenih izzivov v literaturi. Med drugim je varnost podatkov kot večji izziv omenjena v 78 % štu-

dij. Pri tem se s stališča kakovosti podatkov pogosto pojavljajo težave z nekonsistentnostjo podatkov med različnimi komponentami sistema. S stališča IT-arhitekture pa predstavlja večji izziv pravočasnost (angl. timeliness) oz. pravočasno dostopni podatki v določenem koraku cevovoda, ko so ti pričakovani in nujni. Slednje pomeni, da morajo sistemi zagotoviti zelo majhne zakasnitve v vsakem koraku podatkovnega cevovoda, sploh v primeru visokotveganih sistemov (npr. vgrajeni sistemi v avtomobilih).

Naslednji večji izziv v kontekstu sodobnih sistemov velepodatkov je deljenje podatkov, sploh med različnimi oddelki znotraj podjetja. V današnjem okolju več oddelkov znotraj podjetja želi dostopati do iste množice podatkov, ki pa so shranjeni zunaj njihovih IKT-rešitev. V takšnem primeru nastajajo t. i. podatkovni silosi (angl. data silos), kjer ima vsak oddelek svoj interni sistem, ki beleži podatke le 'lokalno', pri čemer se postopek, da se dostop do teh podatkov omogoči tudi zunanjim IKT-rešitvam, lahko močno zaplete. Zaradi tega so zaželene bolj decentralizirane arhitekture, kot je t. i. arhitektura data mesh. V vsakem primeru je treba vzpostaviti učinkovite varnostne mehanizme za nadzor nad deljenjem podatkov, ki jih določajo različne politike, opredeljene kot del upravljanja podatkov.

Za učinkovito deljenje podatkov je treba imeti v mislih tudi optimalno načrtovanje IT-arhitekture glede na potencialna težišča podatkov (angl. data centers of gravity). Težišča podatkov so točke v sistemu, kjer pričakujemo shranjevanje velikih količin podatkov, ki se pozneje premestijo v drugi del sistema (npr. v rešitev za podatkovno analitiko), pri čemer postopek migracije lahko močno vpliva na učinkovitost sistema. Da bi se temu izognili, morajo podatkovni arhitekti pri načrtovanju IT-arhitekture imeti v mislih takšne potencialne točke in oblikovati IT-arhitekturo na način, da je takšnih točk v sistemu čim manj ter da je učinek težnosti podatkov (angl. data gravity) čim manjši. V zvezi s tem se težave pojavljajo tudi zaradi porazdeljenosti okolja oz. podatkov v sistemu [32]. V določenem trenutku je namreč podmnožica podatkov, ki je nujna za izvedbo določene analize, lahko na enem vozlišču v gruči, medtem ko je drugi podnabor na drugem vozlišču. Težava pa nastane, ko je treba v koraku analize uporabiti oba nabora, ki sta logično in velikokrat tudi fizično shranjena na različnih lokacijah v gruči oz. sistemu, in je treba izvesti migracijo potencialno velikih količin podatkov, ne da bi se občutno zmanjšala učinkovitost drugih komponent sistema.

Pri razvoju sistemov za upravljanje velepodatkov se IT-arhitekti in inženirji srečajo s številnimi praktičnimi izzivi, ki nastajajo zaradi lastnosti velepodatkov in hitrega razvoja področja [11]. Eden od izzivov je seveda skaliranje IT-arhitekture, kjer se predvsem usmerimo v horizontalno razširljivost (angl. scale-out). Tukaj nastane izziv, kako določiti optimalno razmerje med kakovostjo podatkov in učinkovitostjo oz. dostopnostjo sistema. Takšna zahteva izhaja tudi iz izbire dveh od treh možnih lastnosti porazdeljenih sistemov, ki ju po teoremu CAP[1] lahko naenkrat zagotovimo. Izbira boljše kakovosti podatkov v sistemu ali učinkovitosti sistema je stvar kompromisa in je zelo odvisna od domenskih zahtev. Naloga IT-arhitektov je načrtovati sistem na način, da poskusijo zagotoviti čim manjše zamude in večjo prepustnost sistema, hkrati pa v določeni meri obdržati konsistentnost podatkov, kolikor je to le možno. Novost področja in nenehno uvajanje novih konceptov, arhitektur in tehnoloških rešitev vpliva na strmo naraščajočo krivuljo učenja pri strokovnjakih, saj je vedno zahtevnejše slediti novostim in najboljšim praksam na področju, tehnološke rešitve pa so vedno bolj zapletene za učenje in implementacijo.

Sodobni sistemi morajo biti zmožni zajemati podatke iz več različnih virov. V tem procesu za integracijo podatkov težave lahko povzroča manjkajoča ustrezna infrastruktura, kar je posebej pomembno v okoljih z različnimi viri podatkov. Slaba implementacija na ravni zajema podatkov predstavlja napačne rezultate analize teh podatkov oz. napačne poslovne odločitve. Hkrati je treba poudariti izziv, da se pri sodobnih IKT-rešitvah, ki so označene kot sistemi velepodatkov, srečujemo tudi z različnimi vrstami podatkov oz. z vedno več pol in nestrukturiranimi podatki (npr. dnevniški zapisi, slike, avdio-video zapisi, prost govor itn.). Nedavno smo se na področju IKT osredotočali zgolj na strukturirane podatke in načine, kako te učinkovito hraniti z uporabo relacijskih modelov ter kako te uporabiti za namen podatkovne analitike, pri čemer smo se kot na orodje za

[1] Teorem CAP (angl. Consistency, Availability, Partition tolerance) pravi, da lahko vsak porazdeljen sistem istočasno zagotavlja le dve od treh možnih lastnosti: celovitost oz. konsistentnost podatkov, razpoložljivost oz. dostopnost sistema ter odpornost sistema na particioniranje.

izvedbo analitike osredotočali na poizvedovalne jezike, kot so SQL, ki so idealni za strukturirane podatke. Nestrukturirane podatke smo rahlo zanemarjali, saj jih nismo znali primerno in učinkovito uporabiti za odkrivanje novih spoznanj. Danes imamo željo in potrebo po tem, da hranimo tudi nestrukturirane podatke, nad katerimi izvajamo podatkovno analitiko, in sicer s pomočjo tehnik podatkovnega rudarjenja, strojnega in globokega učenja itn. Hkrati smo te podatke in to vrsto podatkovne analitike, ki še zmeraj spada pod t. i. OLAP, izvajali zgolj ad-hoc, danes pa so težnje, da se ti podatki in ta vrsta analitike vključi v produkcijske sisteme, ki so povezani s t. i. transakcijsko obdelavo (tj. OLTP). To področje se danes hitro razvija in spada tudi pod t. i. inženiring UI (angl. AI engineering). IT-arhitekti so tako pred kompleksnim izzivom podatkovnega inženirstva, ki je zagotavljati učinkovito podatkovno raven za produkcijske sisteme, večinoma strukturirane podatke in OLTP, ki zahtevajo visoko razširljivost, ter hkrati učinkovito podatkovno raven za analitične procese, nestrukturirane podatke in OLAP, kjer pa se velikokrat srečujemo z masovnimi podatki. Povrh vsega pa se pojavlja težnja za IT-arhitekturo, ki bo omogočala hkratno in učinkovito obdelavo OLTP ter OLAP nad enotno podatkovno ravnjo.

Med trenutnimi izzivi podatkovnega inženirstva so tudi zahteve sodobnih IKT-rešitev in storitev, ki se navezujejo na realnočasovno obdelavo podatkov. Za doseganje ciljev takšnih zahtev se pojavljajo temu primerne podatkovne platforme, kot so platforme sporočilnih sistemov in pretakanja podatkov (npr. Apache Kafka). S tem povezano se tudi pojavljajo vnaprej definirani arhitekturni vzorci, kot sta Lambda in Kappa, ki se osredotočata zgolj na učinkovito obdelavo podatkovnih tokov in pretakanje podatkov [29]. Ker je to precej specifično področje, ki velja zgolj za poslovne primere, kjer je dejanska potreba po realnočasovnem odzivanju na dogodke, se v tem članku na to področje (vele-)podatkovnega inženirstva ne osredotočamo.

Ne nazadnje se sodobni sistemi kot IKT-rešitve srečujejo tudi z nezanemarljivimi stroški vzpostavitve infrastrukture s stališča strojne in programske opreme ter stroški vzdrževanja in človeških virov, saj znajo biti ti zelo visoki ne glede na uporabo oblačne ali lastne infrastrukture (angl. on premise). Takšne sisteme je težko ustrezno nadzorovati, saj je veliko različnih aktivnih komponent v sistemu, nad katerimi je treba v vsakem trenutku imeti popoln nadzor zaradi hitrega odkrivanja okvar v sistemu [11].

3 SODOBNE ARHITEKTURE ZA UPRAVLJANJE VELEPODATKOV

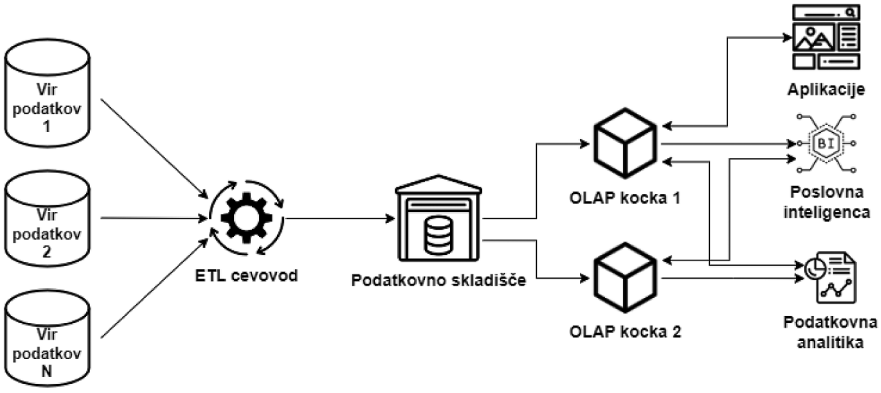
Celovito upravljanje velepodatkov dosežemo z IT-arhitekturo sistema, ki opredeli rešitve in postopke na ravneh zajemanja, shranjevanja, obdelave, vizualizacije in analize velikih količin (potencialno) kompleksnih podatkov [27]. Največja izziva predstavljata zagotavljanje učinkovitega shranjevanja in obdelave velepodatkov, saj zaradi slabega načrtovanja IT-arhitektur nastanejo nemalokrat ozka grla, ki nato vplivajo na celotno učinkovitost sistema. Strokovnjaki si pri načrtovanju podatkovnih cevovodov lahko pomagajo z določenimi smernicami posameznih IT-arhitektur, ki so predstavljene v nadaljevanju.

Za shranjevanje podatkov se uporabljajo različne tehnologije, ki so se razvijale več let in za različne namene. Tako se za shranjevanje transakcijskih podatkov, razen tradicionalnih relacijskih podatkovnih baz, danes pogosto kot zamenjava ali dodatna podpora uporabljajo nerelacijske podatkovne baze (NoSQL), kot sta podatkovna baza ključ-vrednost (npr. Redis) ali dokumentna podatkovna baza (npr. MongoDB), ki rešujejo izzive razširljivosti in prilagodljivosti podatkovne sheme, s katerima se v današnjem okolju srečujejo razvijalci IKT-rešitev. Zahteva večine IKT-rešitev je predvsem podpora za obdelavo OLTP[2], pri čemer je v večini primerov cilj zagotoviti čim večjo konsistentnost podatkov.

Danes imajo večja podjetja več transakcijskih podatkovnih baz oz. podatkovnih zbirk različnih tipov, ki jih je po določenem obdobju treba združiti zaradi izvedbe statističnih analiz in napredne podatkovne analitike, ki so lahko ustrezna podlaga za prihodnje poslovne odločitve, in se podpirajo s pomočjo področja poslovnega obveščanja (angl. business intelligence, BI). Za ta namen je treba ustrezno shraniti združene podatke na eni lokaciji, tj. podatkovno skladišče (angl. data warehouse). Osnova podatkovnih skla-

[1] Teorem CAP (angl. Consistency, Availability, Partition tolerance) pravi, da lahko vsak porazdeljen sistem istočasno zagotavlja le dve od treh možnih lastnosti: celovitost oz. konsistentnost podatkov, razpoložljivost oz. dostopnost sistema ter odpornost sistema na particioniranje.

[2] OLTP (angl. Online Transactional Processing) – način obdelave podatkov kot transakcij, pri čemer se le-te v sistemu shranjujejo in uporabljajo pri obdelavi v realnem času.



Slika 1: **Primer arhitekture z uporabo podatkovnega skladišča.**

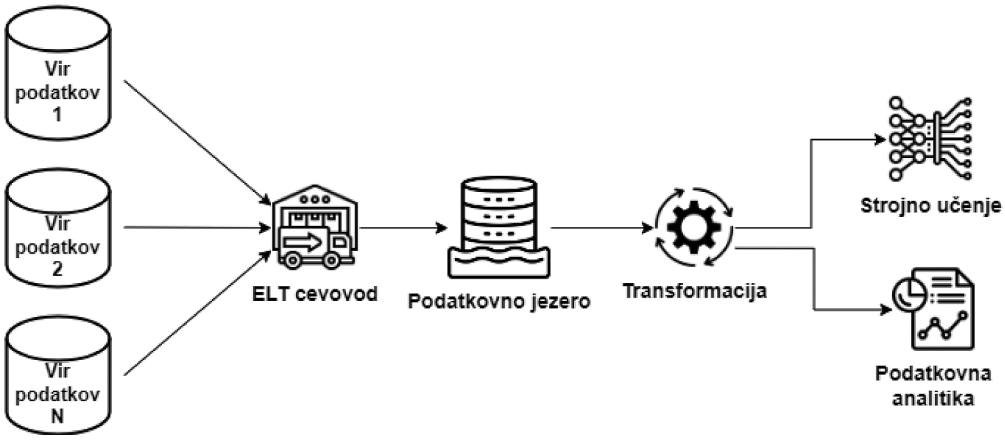
dišč so dimenzijski modeli, ki so v nasprotju z relacijskimi modeli namenski in usmerjeni neposredno v zahteve analitike oz. poslovnega obveščanja. Primer takšne arhitekture je prikazan na sliki 1. Podatki prihajajo iz več podatkovnih virov v cevovod ETL[3], kjer se izvajajo določene transformacije in čiščenje teh v skladu s podatkovno (dimenzijsko) shemo ciljnega podatkovnega skladišča. Prečiščeni podatki se nato shranjujejo v podatkovno skladišče, na podlagi katerega se oblikujejo t. i. kocke OLAP[4] namenjene določenim analizam nad podatki v agregirani obliki. Kocke OLAP vsebujejo podmnožico podatkov, shranjenih v podatkovnem skladišču, in predstavljajo vir podatkov za orodja za poslovno obveščanje ali poročanje. Kot največja izziva pri takšni IT-arhitekturi se poudarjata implementacija učinkovitega procesa ETL in vzpostavljanje mehanizmov za upravljanje kock OLAP v skladu s spremembami pri virih podatkov. Uporabnikom je potem omogočen dostop le do agregiranih podatkov za namene analitike, ne pa tudi izvirnih transakcijskih podatkov.

Zaradi vpliva velepodatkov vlogo podatkovnih skladišč danes prevzemajo širokostolpčne shrambe (angl. wide-column stores), kot sta HBase ali Cassandra, pri katerih so implementirane določene izboljšave za izvajanje analitike (tj. OLAP), ki pozitivno vplivajo na hitrost sistemov in omogočajo shranjevanje transakcijskih ter agregiranih podatkov.

V zadnjem desetletju so se za shranjevanje velepodatkov začela uporabljati tudi podatkovna jezera (angl. data lake) (slika 2), ki omogočajo shranjevanje masovne količine izvirnih (angl. raw) in različno strukturiranih podatkov brez vnaprej določene podatkovne sheme. Kot osrednji repozitoriji izvirnih podatkov so podatkovna jezera primerna za izvedbo zapletenejših analiz z uporabo strojnega učenja. Posledično se najbolj pogosto uporabljajo za podatkovne znanosti, vendar podjetja kot njihovo glavno slabost vidijo odsotnost mehanizmov za zagotavljanje konsistentnosti podatkov, tj. podatkovna jezera ne podpirajo modela ACID kot podatkovna skladišča ali baze[5]. Pri podatkovnih jezerih je cilj čim prej shraniti podatke s pomočjo vzpostavljenega cevovoda ELT[6] cevovoda. Ob branju shranjenih podatkov se podatki transformirajo iz surove oblike (npr. objekti) v obliko, ustrezno za namen uporabe (podatkovne analitike ali strojnega učenja).

Čez leta praktične uporabe se je izkazalo, da današnja podjetja nimajo eksplicitno ločene uporabe transakcijskih od agregiranih podatkov, zaradi česar uporaba le ene predhodno omenjenih tehnologij ne zadostuje, da bi podjetja na optimalen način hkrati zadovoljila potrebe OLTP in OLAP pri upravljanju podatkov. Hkrati je zaradi zahtev po visoki razširljivosti sistema edini primeren način razširjanje navzven (tj. horizontalno), kjer pa so seveda izzivi povezani s teoremom CAP in omejitve relacijskih podatkovnih baz, ki ne podpirajo visoke razširljivosti in hkrati dostopnosti oz. omejitve nerelacijskih podatkovnih baz, ki ne podpirajo visoke razširljivosti in hkrati konsisten-

[3] ETL (angl. Extract-Transform-Load) – proces zajema podatkov iz izvirnega sistema, čiščenja in transformacije prevzetih podatkov glede na ciljni (dimenzijski) podatkovni model ter uvoz pripravljenih podatkov v ciljno shrambo.
[4] Kocka OLAP (angl. Online Analytical Processing cube) – podatkovna struktura, ki omogoča večdimenzijski vpogled v podatke in hitro analizo le-teh.
[5] Model ACID (angl. Atomicity, Consistency, Isolation, Durability) – transakcijski model, s katerim npr. relacijske podatkovne baze zagotavljajo celovitost, konsistentnost, hkratnost in trajnost podatkov v podatkovni bazi.
[6] ELT (angl. Extract-Load-Transform) – proces zbiranja in shranjevanja podatkov v podatkovno jezero, pri čemer se transformacija podatkov izvaja šele ob branju podatkov.



Slika 2: **Primer arhitekture z uporabo podatkovnega jezera.**

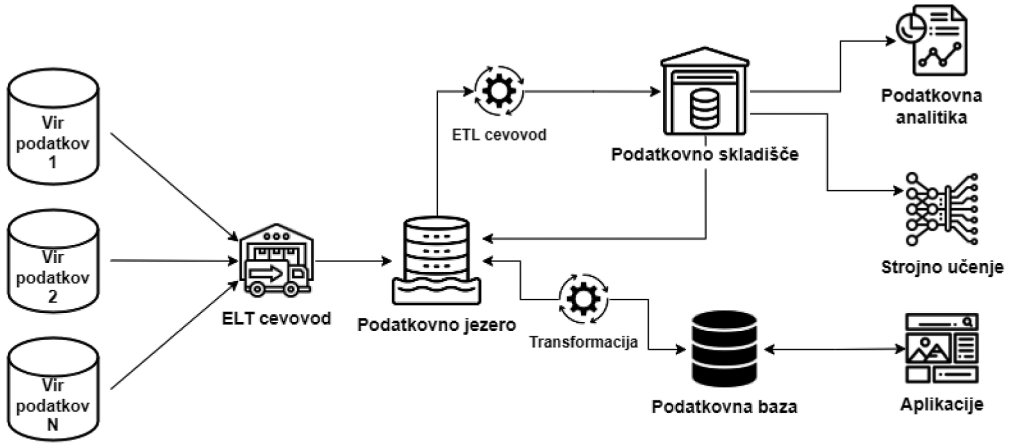
tnosti. Posledično se danes pogosto uporablja dvostopenjska (angl. 2-tier) arhitektura, prikazana na sliki 3, ki vključuje podatkovno jezero in skladišče, s čimer podjetja združijo prednosti obeh tehnologij. V tem primeru se vsi vhodni (transakcijski) podatki shranjujejo v podatkovnem jezeru, del teh pa se agregira in shrani v podatkovno skladišče za potrebe OLAP, od koder se uporabljajo za potrebe poslovnega obveščanja. Vmes pa so vsi podatki shranjeni v izvirni obliki v podatkovnem jezeru in jih lahko podatkovni znanstveniki kadar koli uporabijo za izvedbo zapletenejših algoritmov.

Trenutno je dvostopenjska arhitektura priljubljena izbira podatkovnih arhitektov, saj podjetju omogoča shranjevanje različno strukturiranih podatkov v podatkovnem jezeru, ki so uporabni v sedanjosti in prihodnosti. Dodatni sloj podatkovnega skladišča prinaša podporo za obdelavo podatkov OLAP oz. omogoča izvedbo učinkovitih analiz nad agregiranimi podatki. Podatkovno skladišče se, kot že prej omenjeno, lahko zamenja tudi s širokostolpčno podatkovno bazo. Kot pogosti vzorec se pojavlja tudi vključevanje transakcijske podatkovne baze nad jezerom, s katerimi podjetja lahko izboljšajo učinkovitost obstoječih podatkovnih baz na ravni upravljanja velepodatkov in tudi pohitrijo postopek transformacije surovih podatkov iz jezera za namen podatkovne analitike. V tem primeru je podatkovna baza odložišče podatkov iz jezera, ki so vnaprej transformirani in pripravljeni za analitiko. Uporabljajo se pa tudi IT-arhitekture, kjer pa se transakcijski podatki primarno shranjujejo v relacijsko podatkovno bazo in hkrati tudi v podatkovno jezero.

Čeprav so zajete zahteve z dvostopenjsko arhitekturo, se podjetja srečujejo z določenimi slabostmi takšne implementacije. Največji izziv je zapleteno upravljanje rešitev na obeh stopnjah, kar pomeni, da je treba zagotoviti vire oz. strokovnjake, ki bodo skrbeli, da podatkovno jezero in skladišče oz. tako OLTP kot OLAP delujeta nemoteno. Največ časa pri takšni IT-arhitekturi je treba vložiti v implementacijo zapletenih poslov ETL in ELT, s katerimi se morajo podatki iz podatkovnega jezera transformirati v ustrezni podatkovni model podatkovnega skladišča [2]. Dve ravni shranjevanja pomenita, da se čas izvedbe povpraševanj na strani uporabnikov lahko močno podaljša, saj podatkovna jezera, kakor tudi podatkovna skladišča, imajo določeno mejo, do katere je možno izboljšati učinkovitost izvedbe povpraševanj.

Da bi se izognili potencialno zapletenemu vzdrževanju dvostopenjske arhitekture, so se v novejšem času začela uporabljati t. i. podatkovne hiše ob jezeru ali kolišča[7] (angl. data lakehouses), katerih arhitektura je prikazana na sliki 4. V podatkovnih koliščih so prednosti podatkovnih jezer in skladišč združene znotraj formata oz. modela za shrambo, vendar je dodan sloj metapodatkov, ki vsebuje podrobne opise izvirnih podatkovnih zapisov v obliki objektov, shranjenih v podatkovnem jezeru. Na ta način se znotraj sloja metapodatkov lahko zagotovijo lastnosti ACID-a in dodatni indeksi za izboljšanje konsistentnosti podatkov in učinkovitosti sistema. V tem primeru uporabniki dostopajo do podatkov neposredno iz podatkovnega kolišča, pri čemer se zahtevani podatki najprej poiščejo z uporabo metapodatkov v po-

[7] Podatkovna hiša ob jezeru še nima ustaljenega ali uveljavljenega prevoda v slovenščino, zato uporablja predlog iz islovar.org, tj. kolišče.



Slika 3: **Prikaz dvostopenjske arhitekture za upravljanje velepodatkov.**

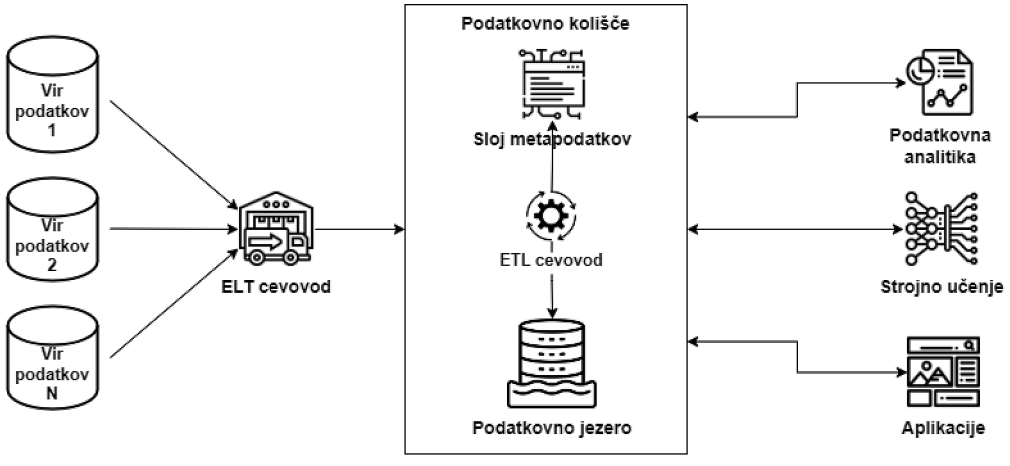
datkovnem jezeru, se transformirajo in/ali agregirajo glede na pravila, shranjena v sloju metapodatkov, se ponovno shranijo na drugo lokacijo v podatkovnem jezeru ter hkrati vrnejo kot rezultat poizvedbe uporabniku. Kot tehnologija so podatkovna kolišča precej nov koncept na trgu, tehnološke rešitve pa se (predvsem za sloj metapodatkov) še vedno razvijajo z vzorci načrtovanja za najboljšo implementacijo arhitekture. Trenutno najbolj uveljavljen primer platforme podatkovnega kolišča je Delta Lake.

Izbira najbolj primerne arhitekture za upravljanje velepodatkov je predvsem odvisna od stopnje strukturiranosti izvirnih podatkov, potreb po prilagodljivosti sheme, konsistentnosti podatkov in razširljivosti ter namenu uporabe shranjenih podatkov oz. ustreznem podatkovnem modelu za določeni namen uporabe (npr. dimenzijski model za analitiko, izvirni podatki za podatkovno znanost, model ključ-vrednost za hitro iskanje ipd.). Izbiro vedno bolj zapletejo novi vzorci načrtovanja in IT-arhitekture, ki želijo združiti prednosti obstoječih rešitev. V nadaljevanju se osredotočimo na analizo tehnoloških rešitev za implementacijo različic dvostopenjske arhitekture, ki je trenutno najbolj priljubljena.

4 ANALIZA IZBRANIH TEHNOLOŠKIH SKLADOV

4.1 Tehnološki sklad Google

Podjetje Google kot eno vodilnih ponudnikov oblačnih storitev v kontekstu upravljanja velepodatkov, ponuja v sklopu Google Cloud Platform precej širok nabor tehnoloških rešitev za implementacijo predhodno predstavljenih arhitektur. Temu v korist je tudi dejstvo, da je za implementacijo dvostopenjske arhitekture pri Googlu možno oblikovati nekaj različic tehnološkega sklada, glede na zahteve strank in željeno stopnjo odvisnosti od ponudnika oblačnih storitev. Ob analizi ponudbe tehnološkega sklada

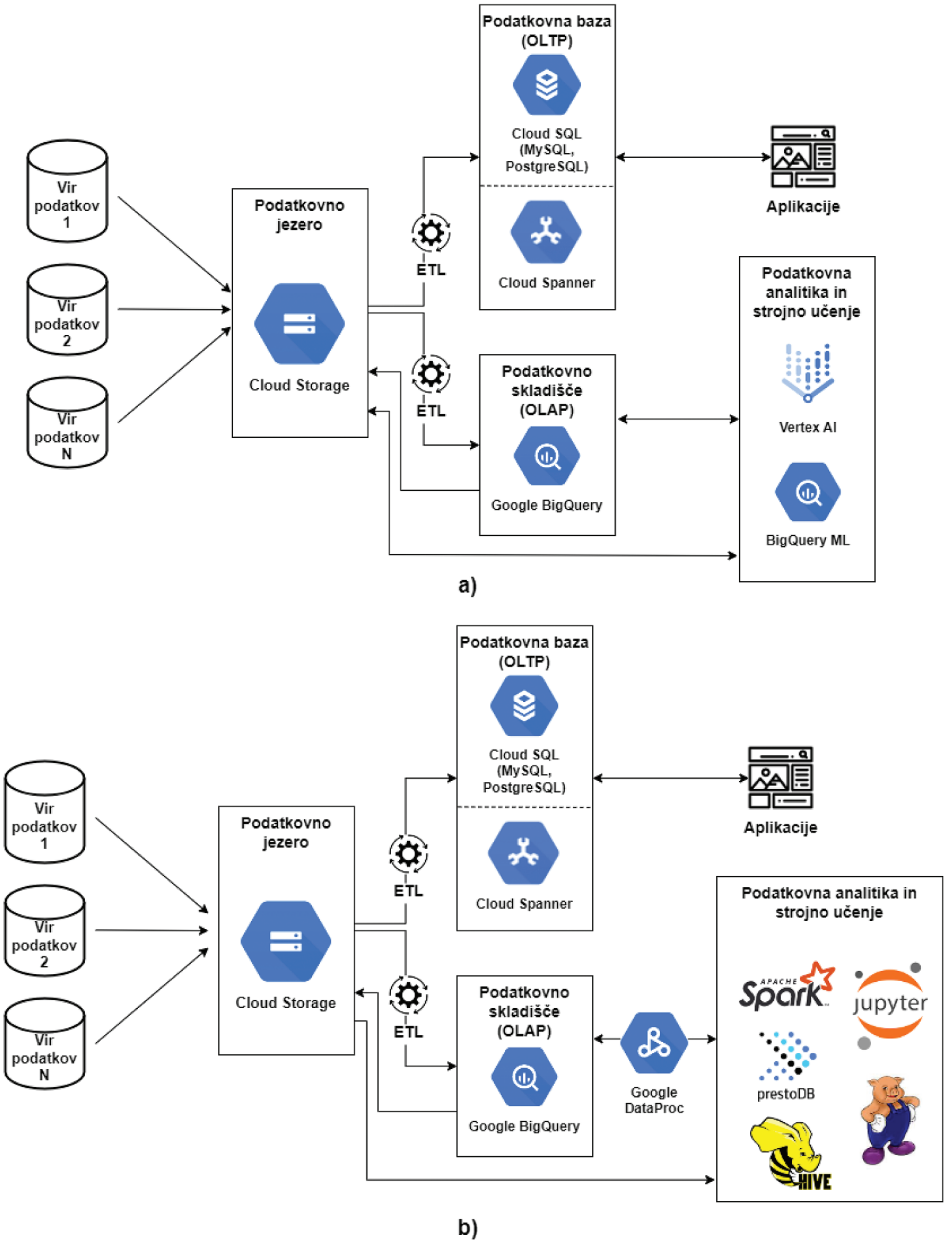


Slika 4: **Prikaz arhitekture z uporabo podatkovnega kolišča.**

Google, smo prišli do pregleda osnovnih rešitev za implementacijo dvostopenjske arhitekture, s katerimi lahko oblikujemo nekaj različic tehnološkega sklada, ki jih predstavljamo v nadaljevanju.

Na ravni podatkovnega jezera se različice tehnološkega sklada ne razlikujejo, saj se za ta namen priporoča uporaba oblačne rešitve za shranjevanje objektov Google Cloud Storage [26]. Cloud Storage je primeren za shranjevanje tudi nestrukturiranih podatkov oz. predstavlja začetno točko shranjevanja izvirnih podatkov. Kot tehnologija temelji na HDFS-u (angl. Hadoop Distributed File System), datotečnem sistemu platforme in ekosistema Hadoop, ki omogoča porazdeljeno shranjevanje velikih količin podatkov v datoteke. Cloud Storage omogoča neomejeno shranjevanje datotek velikosti do pet terabajtov v obliki objektov ter integracijo z različnimi rešitvami za zajem in obdelavo le-teh, pri čemer je optimiziran za prilagodljivo izvedbo povpraševanj in nizke stroške hrambe.

Na sliki 5 sta prikazani dve arhitekturi podatkovnega cevovoda, pri čemer se tehnološki sklad v ozadju razlikuje v izbiri rešitev za podatkovno analitiko



Slika 5: **Prikaz tehnološkega sklada Google z uporabo oblačne relacijske podatkovne baze in različnimi rešitvami za podatkovno analitiko in strojno učenje.**

in strojno učenje. Predstavljena arhitektura cevovoda omogoča podjetjem, da razširijo zmogljivost obstoječih relacijskih podatkovnih baz z vključevanjem podatkovnega skladišča, s čimer implementirajo IT-arhitekturo, ki je zmožna izpolniti analitične potrebe (OLAP) in potrebo po hitri izvedbi neposrednih povpraševanj po relacijski podatkovni bazi (OLTP). Treba je poudariti, da se kot osnova za podatkovno skladišče in analitično obdelavo (OLAP) uporablja širokostolpčna podatkovna baza BigQuery.

V takšni arhitekturi vlogo relacijske podatkovne baze prevzame Google Cloud SQL [25], ki je oblačna storitev za popolno upravljanje relacijske baze (trenutno so podprte MySQL, PostgreSQL in SQL Server). Storitev se lahko integrira v obstoječe IS-okolje podjetja s pomočjo storitev za orkestracijo, kot so App Engine, Google Kubernetes ipd., ter obstoječimi aplikacijami in storitvami, kot je Google BigQuery. Integracija z rešitvijo BigQuery omogoča vpeljavo analitičnih možnosti bodisi neposredno nad komponento BigQuery Storage bodisi nad podatkovno bazo Cloud SQL, kar lahko močno izboljša učinkovitost dostopa do podatkov.

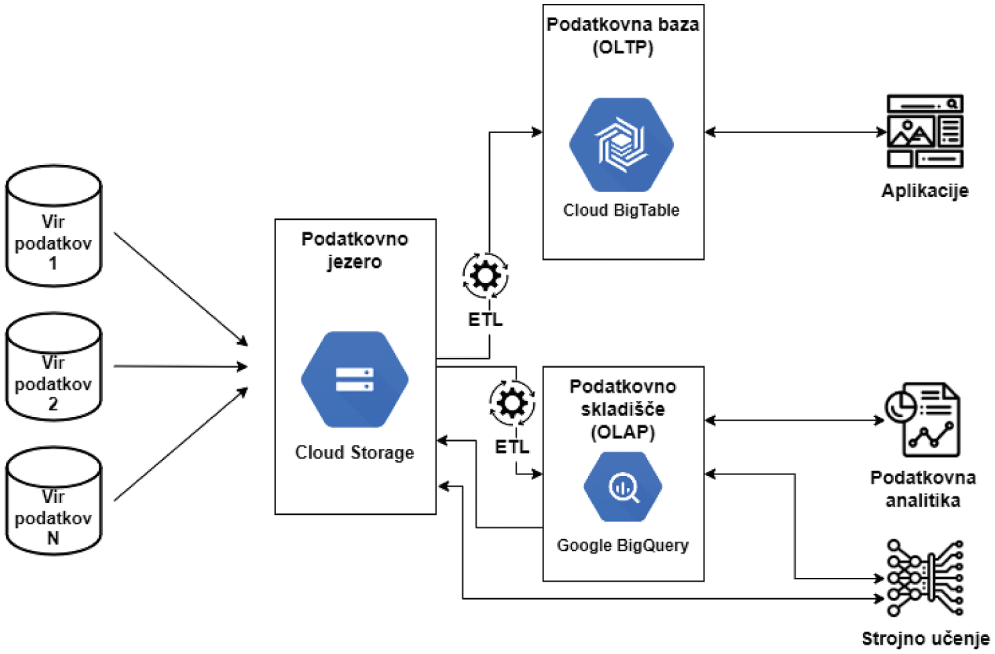
Če podjetje želi svoj transakcijski del (OLTP) učinkoviteje skalirati, se lahko uporabi rešitev Google Cloud Spanner za upravljanje relacijskih podatkovnih baz [24]. To je podatkovna baza, ki zagotavlja lastnosti ACID-a, a omogoča podprto horizontalno razširljivost, samodejno replikacijo podatkov, črepinjenje (angl. sharding) in obdelavo transakcij.

Privzeto je nastavljen Cloud Spanner, ki odvisno od količine podatkov in obremenitve sistema samodejno izvaja črepinjenje podatkov s ciljem izboljšanja učinkovitosti, vendar ni namenjen za uporabo v primerih, kjer je treba zadovoljiti osnovne potrebe SQL oz. OLTP (npr. osnovna analitika nad manjšo količino podatkov). Kot rešitev je Cloud Spanner uporabnejši v primerih, ko se pričakujeta masovna količina podatkov s pogostim pisanjem v podatkovno bazo (npr. tisoče operacij pisanja v sekundi) in visoka razširljivost sistema OLTP.

Po drugi strani se v kontekstu implementacije podatkovnega skladišča oz. sistema OLAP v tehnološkem skladu Google najbolj pogosto omenja Google BigQuery [20] – rešitev, ki je že od začetka predstavljala kritično točko prehoda v ekosistem velepodatkov in predstavlja navdih številnim sodobnim rešitvam za upravljanje velepodatkov. BigQuery je implementacija podatkovnega skladišča v več oblakih (angl. multicloud), kar omogoča analizo podatkov, shranjenih v več oblačnih storitvah. V osnovi to ni tipično podatkovno skladišče, ampak stolpčna shramba s strojem za izvedbo povpraševanj (angl. query engine), optimiziranim za analitične potrebe oz. hitre operacije branja. Predstavlja ustrezno rešitev poizvedbam, ki zahtevajo skeniranja celih tabel in izvedbo operacij grupiranja podatkov (npr. iskanje povprečja). Uporabnikom je rešitev BigQuery prijazna, saj tam lahko shranijo podatke, ki imajo strukturo relacijske tabele, dodatno pa vključuje nabor orodij za podatkovno analitiko, s katerim se že lahko neposredno ustvarijo nadzorne plošče (angl. dashboards) in generirajo poročila. Za namen analize podatkov BigQuery vključuje naslednji nabor orodij:

- BigQuery ML (angl. Machine Learning) – nabor orodij namenjen podatkovnim znanstvenikom in analitikom za izgradnjo in vzpostavitev modelov za strojno učenje z uporabo sintakse SQL (angl. Structured Query Language) na velikih količinah različno strukturiranih podatkov, shranjenih neposredno v BigQuery-u;
- BigQuery BI Engine (angl. Business Intelligence) – servis za podatkovno analitiko v pomnilniku (angl. in-memory), vgrajen v BigQuery, ki omogoča interaktivno analizo velikih in kompleksnih naborov podatkov, pri čemer se ohranjajo učinkovitost in hitrost izvedbe povpraševanj ter visoka konkurenčnost.

Google BigQuery je popularna rešitev s širokim naborom možnosti uporabe samo za shrambo in/ali obdelavo podatkov. Posledično je tudi model plačevanja storitve ločen na plačevanje rešitve izključno za namen učinkovitega shranjevanja podatkov in plačevanje za namen uporabe orodij za analitiko. BigQuery je priljubljen tudi zaradi podpore za združevanje zmogljivosti podatkovnega skladišča in podatkovnega jezera brez potrebe po uporabi tretje rešitve. Na ravni analize podatkov pa se pri uporabi rešitve BigQuery tehnološki sklad Google lahko oblikuje na različne načine. Kot primer sta na sliki 5 prikazani dve arhitekturi. Pri arhitekturi na sliki 5a se za podatkovno analitiko in strojno učenje uporabljajo orodja vgrajena v ekosistem Google (BigQuery), pri čemer uporabniki plačajo stroške shranjevanja in obdelave v rešitvi BigQuery. Kot alternativa za znižanje stroškov uporabe Google Cloud Platforme (BigQuery) se lahko vzpostavi tudi arhitektura, prikazana na sliki 5b, kjer se za analizo podatkov uporabljajo odprtokodne



Slika 6: **Prikaz tehnološkega sklada Google z uporabo oblačne nerela**cijske podatkovne baze.

rešitve, kot so Apache Spark, Presto, Hive, Pig. Na ta način uporabniki poravnajo le strošek shranjevanja podatkov v BigQueryju, za obdelavo podatkov pa se lahko v BigQuery integrirajo zgornje rešitve znotraj ekosistema Apache. Slednja integracija je možna z uporabo orodja Google DataProc [22], s katero te rešitve lahko pišejo ali berejo podatke neposredno v/iz baze BigQuery s pomočjo BigQuery Storage API-ja.

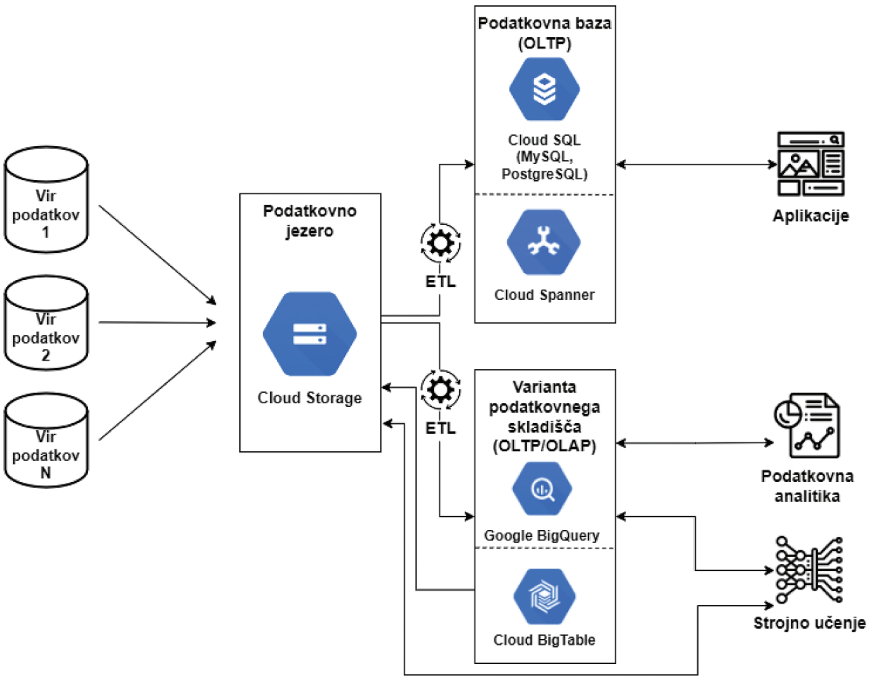
Če podjetje želi zamenjati obstoječo relacijsko bazo zaradi morebitnih izzivov in pomanjkljivosti, kot sta razširljivost in neprilagodljivost sheme, se podjetje lahko odloči tudi za uporabo nerelacijske baze za del OLTP, pri čemer nastane različica arhitekture, prikazana na sliki 6.

V tehnološkem skladu Google se za ta namen uporablja Google BigTable [21], ki je razširljiva storitev NoSQL za uspešno upravljanje in izvedbo velikih analitičnih in operativnih delovnih obremenitev (angl. workload). Podpira model sčasome konsistentnosti (angl. eventual consistency), kar pomeni, da se podatki v bazo shranijo enkrat in se samodejno replicirajo na vozlišča po potrebi. Kot baza BigTable podpira visoko prepustnost za operacije pisanja in branja z nizko zakasnitvijo in predstavlja idealen vir podatkov za posle MapReduce, saj je zaledni stroj za hrambo (angl. storage engine) načrtovan v smeri, ki je primeren za lažje strojno učenje in napredno analitiko.

Podatkovni model baze Google BigTable je kombinacija shranjevanja podatkov v obliki ključ-vrednost in delno širokostolpčne shrambe. Podatki se namreč shranjujejo v visoko razširljivih tabelah, vsaka tabela pa se predstavlja kot razvrščena mapa ključev in vrednosti. Zaradi slednjega je oblika shrambe BigTable primerna tudi za izvedbo nalog OLAP, saj omogoča hitro iteracijo po ključih. Po drugi strani se lahko pri oblikovanju uporabljajo tudi družine stolpcev (angl. column families), s čimer imajo uporabniki dostop do določenih prednosti širokostolpčnih shramb. Koncept družine stolpcev je pozneje razširjen v izjemno razširjenih široko stolpčnih bazah, kot sta Apache HBase ali Cassandra, ki so nastale na podlagi BigTabla.

Pri izvedbi povpraševanj BigTable ne podpira poizvedovalnega jezika SQL, vendar se lahko integrira z rešitvijo Google BigQuery, pri čemer nastane različica arhitekture, prikazana na sliki 7. Integracija BigTable in rešitev BigQuery predstavljata rešitev, ki lahko prevzame vlogo podatkovnega skladišča, saj zmore obvladati naloge OLTP in OLAP. V tem primeru nerelacijska baza, kot je BigTable, shranjuje transakcijske podatke v obliki, ki je vnaprej optimizirana za izvedbo povpraševanj, medtem ko močna rešitev, kot je BigQuery, lahko bere podatke in izvaja osnovne in napredne analize in agregacije.

Včasih se lahko zgodi, da uporaba relacijske podatkovne baze kot sistema OLTP ne zadostuje potrebam podjetja, saj imajo relacijske baze določene znane



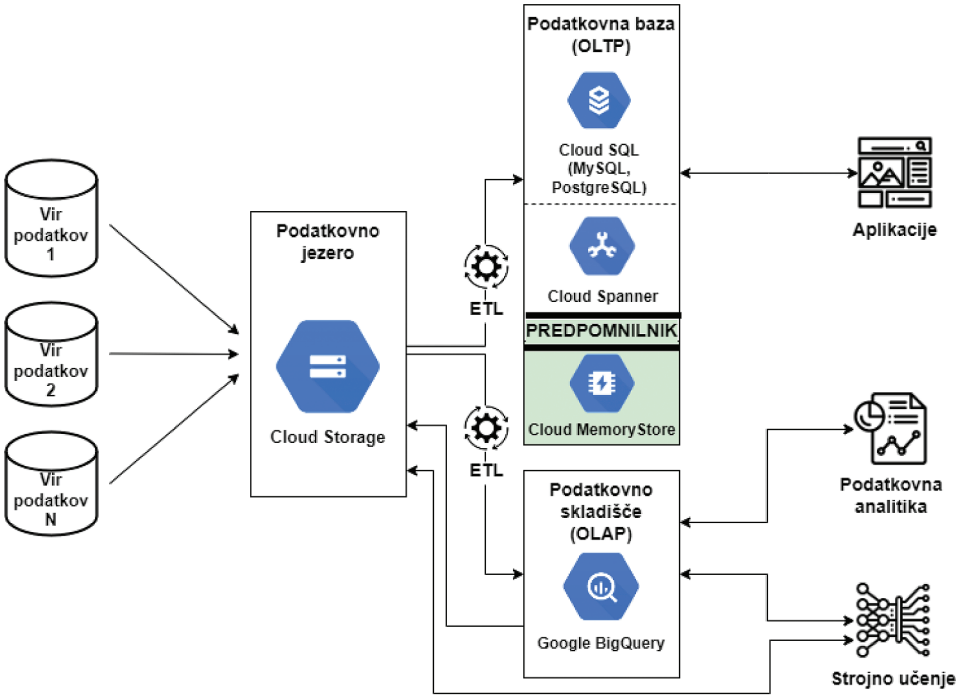
Slika 7: **Prikaz tehnološkega sklada Google z uporabo različice podatkovnega skladišča OLTP/OLAP.**

pomanjkljivosti v kontekstu velepodatkov (npr. razširljivost, prilagodljivost sheme). Ne glede na izbrano različico arhitektur, ki so predhodno predstavljene, se lahko zgodi, da podjetje ni zadovoljno z učinkovitostjo, ki jo v osnovi ponuja relacijska baza (tudi z ukrepi za optimizacije in izboljšave) ali pa je obremenitev sistema OLTP preprosto previsoka. V takšnem primeru se priporoča uporaba določene rešitve, ki temelji na dostopu do podatkov znotraj delovnega pomnilnika, saj se na ta način močno zmanjša število dostopov do diska, kar je posledica hitrejših poizvedb. Za ta namen Google ponuja uporabo storitve v delovnem pomnilniku Cloud Memorystore (slika 8). Rešitev je popolnoma kompatibilna z odprtokodnimi rešitvami, kot sta Redis in Memcached, ki se tudi uporabljajo za izboljšanje učinkovitosti sistema OLTP [23].

4.2 Tehnološki sklad Amazon

Tehnološki sklad Amazon vključuje ožji nabor rešitev, ki se sicer lahko uporabijo za več različic pri implementaciji dvostopenjske arhitekture. Kot je prikazano na sliki 9, vlogo podatkovnega jezera pri skladu Amazon prevzame Amazon S3 [15], dobro znana rešitev, ki se pogosto uporablja kot osnovna rešitev za shranjevanje podatkov v obliki datotek (angl. file server). S3 je največja in ena najbolj učinkovitih storitev za shranjevanje objektov za strukturirane in nestrukturirane podatke, vendar v novejšem času postaja tudi priljubljena rešitev za vzpostavitev podatkovnega jezera. Eden od razlogov za to je, da storitev samodejno ustvarja in shranjuje kopije vseh vnesenih objektov S3 v več porazdeljenih vozliščih. Kot razlog za uporabo rešitve S3 se poudarja tudi močna razširljivost na ravni petabajtov, ki omogoča navidezno neomejeno shranjevanje podatkov v kakršni koli obliki. Osnovni princip v ozadju je porazdelitev med hrambo in obdelavo podatkov, kar prinaša več prednosti predvsem pri vzdrževanju in uporabi virov. Glede na pogostost uporabe in dostopanja do različnih naborov podatkov, se uporabniki lahko odločijo za nakup različic storitev S3 Standard ali S3 One Zone Infrequent Access [18]. Prva je splošna različica za shranjevanje podatkov, ki jo pogosto uporabljamo pogosto za različne namene, medtem ko je različica Infrequent Access ustrezna izbira za shranjevanje dolgotrajnih podatkov, do katerih dostopamo redko.

Pri uporabi rešitve S3 kot podatkovnega jezera je treba vzpostaviti različna območja (angl. zones) [30, 12] oz. sloje za shranjevanje podatkov kot objektov glede na različne stopnje obdelave in korak v podatkovnem cevovodu. Tako imenovano območje landing predstavlja izhodiščno točko shranjevanja prispelih podatkov iz virov podatkov v izvirni obliki v cevovodu (npr. JSON, XML, CSV), preden se začnejo obde-



Slika 8: **Prikaz tehnološkega sklada Google z uporabo predpomnilnika.**

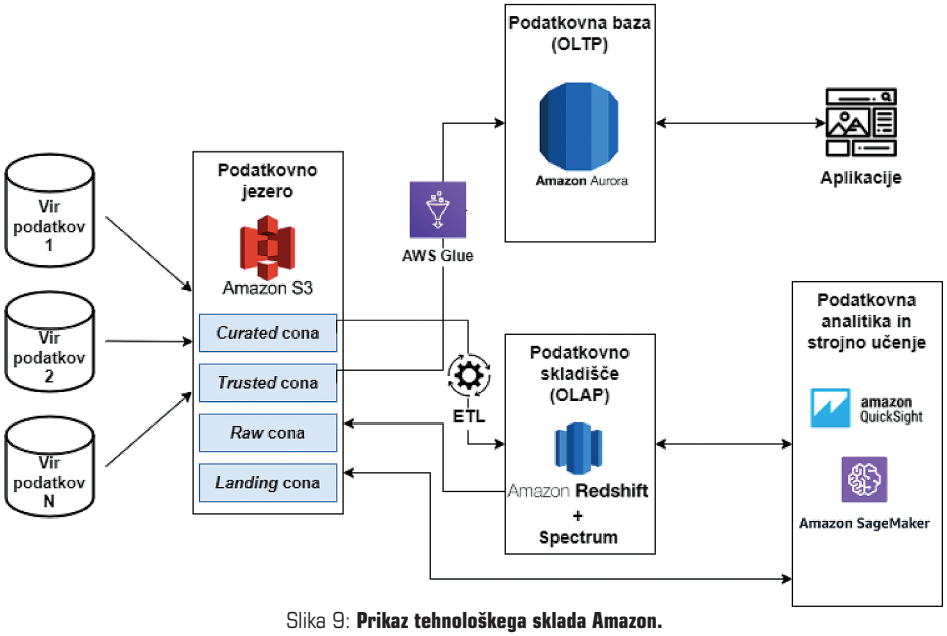
lovati na višjih ravneh. Podatki iz območja landing se validirajo in ustrezno reorganizirajo ter shranijo v t. i. območju raw, ki še vedno vsebuje podatke v izvirni obliki, vendar je struktura oz. organizacija objektov izboljšana glede na potrebe, podatki pa se shranijo v formatu, ki zagotavlja boljšo učinkovitost nadaljnjih analiz (npr. Parquet).

Izvirni podatki se nato transformirajo v skladu z določeno podatkovno shemo in hranijo v t. i. območju trusted. Do območja trusted lahko dostopa kateri koli sistem OLTP oz. podatkovna baza, saj so podatki konsistentni, prečiščeni in usklajeni z določeno shemo. Na najvišji ravni se podatki v območju trusted združujejo glede na potrebe analiz in se združeni hranijo v t. i. območju curated. Ta je tako glavni vir podatkov za podatkovno skladišče, kjer se podatki agregirajo za potrebe analitike in prikaza uporabniku.

Zaradi zahtevnejše organizacije podatkov znotraj shrambe S3 se priporoča vzpostavitev mehanizma za samodejni zajem podatkov iz virov, ustvarjanje in vzdrževanje kataloga metapodatkov ter zagotavljanje točnosti podatkovnih tokov od različnih območij. Za ta namen se priporoča uporaba rešitve AWS Glue [14], ki je popolnoma upravljana storitev za ETL, ki lahko olajša procese razvrščanja, čiščenja, transformacije in prenosa podatkov med različnimi lokacijami. AWS Glue Data Catalogue [17] je orodje znotraj rešitve AWS Glue, ki zagotavlja enotni repozitorij metapodatkov za izvedbo operacij za namen analitike nad več različnimi viri podatkov, kot so Amazon EMR, Athena, Redshift (Spectrum) ali katere koli aplikacije kompatibilne s hrambo Hive za metapodatke (angl. Hive metastore). Data Catalogue je predvsem podatkovna baza, v katero se shranjujejo metapodatki in tabele podatkovnih shem, lokacija podatkov v sistemu in različne metrike za delovanje rešitve oz. sistema. Ker je kompatibilen s hrambo Hive za metapodatke, pa se Data Catalogue lahko uporablja kot osrednji repozitorij za shranjevanje strukturiranih in nestrukturiranih metapodatkov.

Ko so podatki shranjeni v S3 se v naslednjem koraku cevovoda transformirajo skozi procese ETL, ki postavijo le tiste v ustrezno obliko za potrebe analitike ali vizualizacije. Pri transformaciji podatkov se lahko uporabijo trije pristopi [14]:

- Ustvarjanje gruče Amazon EMR z nameščeno rešitvijo Hive – surovi podatki, shranjeni v S3, se transformirajo v tabele Hive in se shranijo v S3 v formatu Parquet.
- Uporabljanje rešitve Spark na Amazon EMR.
- Uporabljanje rešitve AWS Glue, ki samodejno najde surove podatke, shranjene v S3, identificira njihov format shrambe ter predlaga ciljno shemo in transformacije.



Slika 9: **Prikaz tehnološkega sklada Amazon.**

V tehnološkem skladu Amazon se kot sistem OLTP oz. transakcijska podatkovna baza uporablja Amazon Aurora [13], storitev za implementacijo relacijske baze, ki združuje hitrost in dostopnost komercialnih podatkovnih baz s preprostostjo in stroškovno učinkovitostjo odprtokodnih podatkovnih baz. Aurora je popolnoma kompatibilna s sistemi za upravljanje podatkovnih baz MySQL in PostgreSQL, kar bistveno olajša integracijo z obstoječimi aplikacijami, ne da bi bile nujne velike spremembe. Prilagodljiva je vsem potrebam po razširljivosti, saj se viri shrambo po potrebi samodejno razširijo. Poslovni model vzpostavitve rešitve Aurora vključuje plačilo storitve po uri uporabe posamezne instance Aurore.

Na drugi strani sklada se kot glavna rešitev za OLAP uporablja Amazon RedShift [16] – hitro podatkovno skladišče na ravni petabajtov, s katerim lahko uporabniki analizirajo svoje podatke, shranjene na več različnih lokacijah. Na ravni implementacije je RedShift širokostolpčna podatkovna baza, ki jo je z uporabo širokega nabora vtičnikov možno povezati z različnimi odjemalci oz. bazami, kot je PostgreSQL. Rešitev je preprosto razširljiva, saj je po potrebi možno dodati nova vozlišča v oblačno storitev s pomočjo spletne konzole ali zasebnega API-ja. Vozlišča lahko dosežejo kapaciteto med 160 gigabajtov in 16 terabajtov. Uporabniki poravnajo dejansko porabo virov. RedShift predstavlja izjemno močno rešitev znotraj sklada Amazon, s katero lahko podjetja zadovoljijo analitične potrebe, dodatna prednost pa je tudi možnost povezovanja RedShifta z relacijsko podatkovno bazo ali rešitvijo za poslovno obveščanje zunaj ekosistema Amazon. Za delo z RedShiftom se uporablja robusten API, ki omogoča izvedbo poizvedb nad podatki shranjeni v bazi. Za dodatno optimizacijo uporabe virov ob izvedbi povpraševanj RedShift uporablja algoritme strojnega učenja za napovedovanje in analizo povpraševanj. RedShift podpira različne formate podatkov v obliki datotek, kot so Parquet ali ORC (angl. Optimized Row Columnar), ki lahko vplivajo na učinkovitost sistema. Kot omejitev uporabe RedShifta se omenjata OLAP, omejitve, povezane z učinkovitostjo operacij vnašanja, posodabljanja in brisanja ter izostanek mehanizma za upravljanje indeksov znotraj platforme AWS.

Znotraj rešitve RedShift obstaja tudi orodje, ki omogoča hitro izvedbo kompleksnih analiz nad objekti, shranjenih v določenih oblačnih rešitev sklada Amazon (S3, RedShift itn.). Amazon RedShift Spectrum [19] je orodje, ki se pogosto uporablja z RedShiftom, saj omogoča samodejno skaliranje in optimalno izvedbo povpraševanj (na ravni eksabajtov podatkov) nad gručo RedShift. Uporaba orodja se poravna tudi po porabi, nujno pa potrebuje vzpostavljeno RedShift gručo in povezanega odjemalca SQL. Z uporabo znane sintakse jezika SQL znotraj orodja Spectrum lahko uporabniki hitro dostopajo do velikih količin podatkov porazdeljenih med več gruč RedShift (ali vozlišč) in izvajajo kompleksna povpraševanja nad podatki. Rezultati povpraševanj se lahko

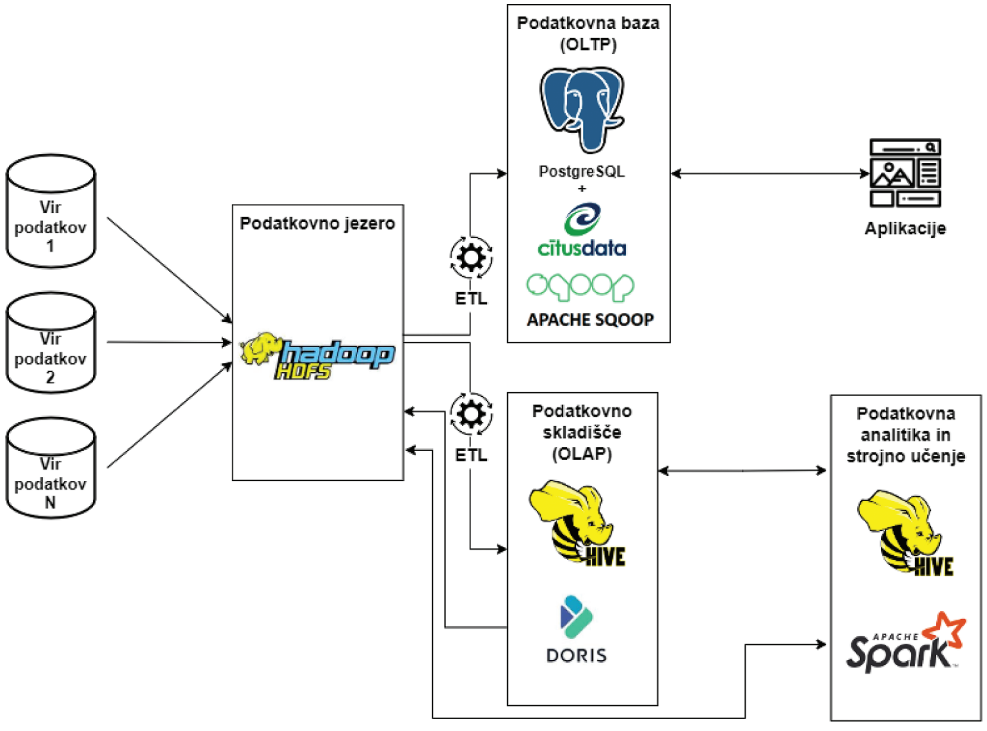
nato uporabijo za namen podatkovne analitike, ki jo znotraj sklada Amazon izvajamo z orodjem Amazon QuickSight, ali strojnega učenja, za katero potrebujemo rešitev Amazon SageMaker.

4.3 Odprtokodni tehnološki sklad Apache

Kot alternativa plačljivim storitvam in rešitvam, ki jih ponujajo vodeči ponudniki na trgu, kot so Google (Google Cloud Platform – GCP), Amazon (AWS) in Microsoft (Azure), se podjetja lahko obrnejo tudi proti odprtokodnim tehnologijam, kar je tudi prednostna smer implementacije pri večini podjetij z omejenim proračunom za implementacijo sistema. Pri tem se kot sopomenka za odprtokodne rešitve na področju velepodatkov večinoma uporablja Apache, saj predstavlja nabor programske opreme, ki je na voljo vsem uporabnikom. Rešitve, ki jih ponuja sklad Apache, so dovolj hitre in zanesljive za uporabo ter se lahko prilagodijo posameznim potrebam podjetja, kar je največja dodana vrednost za podjetja, ki imajo specifične poslovne procese ali zahteve in želijo imeti večji nadzor nad implementacijo. Čeprav implementacija sklada Apache zahteva več časa in znanja, brezplačna uporaba rešitev za upravljanje velepodatkov, ki niso t. i. 'črna škatla' (angl. black box), podjetju predstavlja upravičen strošek za uporabo prav tistega sklada namesto plačevanja oblačnih storitev pri predhodno omenjenih ponudnikih.

Čeprav je nabor možnih rešitev za posamezo komponento dvostopenjske arhitekture pri skladu Apache precej širši kot pri skladih Google in Amazon, je v nadaljevanju na sliki 10 predstavljen osnovni predlog sklada z rešitvami, ki predstavljajo jedro tehnoloških skladov za upravljanje velepodatkov.

Za funkcionalnosti podatkovnega jezera pri skladu Apache poskrbi rešitev HDFS (angl. Hadoop Distributed File System) [8], porazdeljeni datotečni sistem, ki predstavlja osnovno lokacijo za shranjevanje podatkov v obliki datotek znotraj platforme in ekosistema Hadoop. Podatki se znotraj HDFSa shranujejo kot bloki, pri čemer se vsak blok za namen replikacije razdeli trikrat na različna HDFS podatkovna vozlišča. HDFS predstavlja hrbtenico ekosistema Hadoop in podobnih datotečnih sistemov drugih ponudnikov na trgu, kot so Microsoft HDInsight, Cloudera CDH, IBM Open Platform itn. Kot alternativa uporabi HDFS-a kot podatkovnega jezera so dandanes na trgu dostopne tudi rešitve zunaj sklada Apache, in sicer se lahko uporabi npr. MinIO kot oblačna rešitev visoke zmogljivosti, ki je hkrati kompatibilna z rešitvijo Amazon S3. HDFS deluje po principu, da vsaka naprava v gruči Hadoop hrani podmnožico podatkov,



Slika 10: **Prikaz tehnološkega sklada Apache.**

ki tvorijo datotečni sistem. Ob potrebi po shranjevanju več podatkov se lahko v gručo doda več naprav z več diski in se celotni datotečni sistem preprosto razširi. Podatki, shranjeni v HDFSu, se nato posredujejo v nabor poslov ETL, ki jih lahko opravimo z Apache Sparkom ali katerim koli orodjem, ki omogoča uvoz in izvoz podatkov v/iz HDFSa. Primer takšnega orodja je Apache Sqoop [5], ki omogoča prenos paketnih podatkov med gručo Hadoop in strukturiranimi viri podatkov, kot so relacijske baze ali skladišča.

Z uporabo Sqoopa se podatki iz HDFSa lahko transformirajo in shranijo v relacijsko bazo, kot je PostgreSQL, ki nato prevzame vlogo sistema OLTP v celotni arhitekturi. Za izboljšano učinkovitost in razširljivost klasične baze PostgreSQL se uporablja odprtokodna razširitev Citus [4], ki vpeljuje možnost porazdeljenega shranjevanja in obdelave podatkov v bazi PostgreSQL in učinkovito razširi zmogljivost obstoječe podatkovne baze PostgreSQL. Z vpeljavo rešitve Citus lahko podjetja razširijo obstoječo podatkovno bazo PostgreSQL na gručo vozlišč PostgreSQL, ki so zmožna učinkovito hraniti več podatkov. Nastane gruča Citus, v kateri eno vozlišče predstavlja koordinatorja Citus, ki posreduje povpraševanja, napisana v jeziku SQL, na ustrezno vozlišče PostgreSQL in vodi evidenco o lokaciji, kjer je določeni podatek v gruči.

Vlogo podatkovnega skladišča v skladu Apache lahko prevzame rešitev Apache Doris [6], podatkovno skladišče OLAP, ki omogoča uporabo poizvedovalnega jezika SQL in temelji na principu masovne vzporedne obdelave (angl. massively parallel processing). To je relativno nova rešitev, razvita v inkubatorju Apache, ki je nastala kot rezultat integracije rešitev Apache Impala (porazdeljeni stroj SQL za povpraševanja nad gručo Hadoop) in Google Mesa (visoko razširljivo podatkovno skladišče, razvito s strani Googla). Doris omogoča preprosti načrt IT--arhitekture, ki hkrati zagotavlja visoko zanesljivost, dostopnost, razširljivost ter toleranco na napake. V ozadju rešitve Doris je širokostolpčna podatkovna baza, vendar je le-tista nadgrajena s tehnologijo vektorizacije (angl. vectorization technology) zaradi optimizacije pri izvedbi povpraševanj. Tradicionalni stroji SQL za povpraševanja (angl. SQL query engines) analizirajo podatke v tabelah po principu vrstic (angl. row- based), kar prinaša dodatni strošek pri uporabi procesorskih enot CPU. Pri Doris želi tehnologija vektorizacije izboljšati pristop na način, da se podatki obdelajo po principu stolpcev, kar zmanjšuje uporabo virov CPU in v celoti eliminira odvečno število operacij branja nad podatki. Doris podpira tudi visoko konkurenčnost dostopa med več uporabniki ter horizontalno razširljivost. Uporabniki lahko dostopajo do podatkov, shranjenih v skladišču, z različnimi odjemalci in orodji za poslovno obveščanje. Kot največjo prednost rešitve poudarjajo možnost realnočasovnih nadzornih plošč, ad hoc poizvedbe in celovito rešitev za razširljivo podatkovno skladišče, ki lahko zagotovi funkcionalnosti, ki bi jih sicer mogli nadomeščati z več rešitvami (Spark, Hive in HBase). Kot alternativa Apache Doris je vsekakor tudi Apache HBase, ki je ena pravih odprtokodnih rešitev širokostolpčne podatkovne baze.

Kot pomembna komponenta sklada Apache se lahko kljub temu uporablja Hive [7] – podprt sistem SQL za analiziranje podatkov, shranjenih v HDFSu. Zaradi preprostega jezika za pisanje poizvedb (Hive Query Language, HQL) je Hive ustrezna rešitev za obdelavo podatkov in izvedbo procesa ETL, zaradi neposrednega dostopa do podatkov v HDFSu pa se lahko uporablja tudi kot alternativa za podatkovno skladišče. V primerjavi z drugimi rešitvami sklada Apache (npr. Impala ali Doris) je bolj namenjen podatkovnim inženirjem in ne podatkovnim znanstvenikom. V tistem primeru se podatki, shranjeni v podatkovnem jezeru oz. HDFSu, indeksirajo v komponenti Hive MetaStore, ki deluje kot katalog metapodatkov, v katerem se struktura datotek HDFS mapira v obliko razpredelnice. Hive je zgrajen nad Hadoopom, kar pomeni, da je že v načrtu ustrezna rešitev za potrebe upravljanja velikih datotek na ravni petabajtov. Kot rešitev omogoča sloj abstrakcije nad HDFSom, na katerem so podatki predstavljeni kot tabele z vrsticami, stolpci in podatkovnimi tipi, katere uporabniki lahko analizirajo po vmesniku HiveQL. Kot dodatna prednost je podpora za transakcije ACID, kar pomaga pri zagotavljanju konsistentnosti podatkov. Hive sledi pristopu sheme ob branju (angl. schema-on- read), kar pomeni, da se podatki hitro shranjujejo v skladišče Hive brez validacije ali dodatnih preverjanj, končno obliko pa pridobijo iz Hive MetaStora ob izvedbi povpraševanj. Glede integracije z drugimi rešitvami, kot je Amazon S3, je Hive dovolj odprta rešitev. Razen shranjevanja združenih podatkov v skladišču se Hive lahko uporablja tudi na ravni podatkovne analitike v primeru osnovnih analiz. Za naprednejšo analitiko in strojno učenje se priporoča uporaba rešitve Apache Spark, ki predstavlja danes najbolj razširjeno rešitev za obdelavo ve-

lepodatkov [31] zgrajeno nad porazdeljenim strojem SQL za izvedbo povpraševanj nad velepodatki.

5 DISKUSIJA

IT-arhitekti sodobnih podatkovnih cevovodov imajo izziv izbrati tehnološki sklad za implementacijo dvostopenjske arhitekture, ki je v skladu z dostopnimi viri ter omejitvami in zahtevami vodstva podjetja. Čeprav ponudniki, kot sta Google ali Amazon, težijo k plačljivemu načinu uporabe njihovih rešitev, je možno določene rešitve znotraj njihovega sklada integrirati z odprtokodnimi rešitvami znotraj sklada Apache ali drugega sklada, s katerima si podjetja lahko znižajo stroške implementacije. V Tabeli 1 so predstavljeni izsledki analize rešitev tehnoloških skladov Google, Amazon in Apache, ki so arhitektom na voljo za implementacijo posameznih komponent dvostopenjske arhitekture, in sicer možne rešitve za podatkovno jezero, podatkovno skladišče (OLAP), podatkovno bazo (OLTP) ter rešitve za podatkovno analitiko in strojno učenje. Poudariti pa je treba, da IT-arhitekti niso zgolj omejeni z uporabo odprtokodnih rešitev za podatkovno analitiko, kot prikazuje slika 5b, temveč lahko posamezne komponente (npr. podatkovno jezero, podatkovno skladišče, podatkovna baza) izbirajo s strani različnih ponudnikov in te med seboj združujejo. Primer tega bi bila uporaba Amazon S3 za podatkovno jezero, relacijsko podatkovno bazo PostgreSQL na lastni infrastrukturi za namen OLTP, Google BigQuery kot osnova za podatkovno skladišče oz. OLAP ter odprtokodna orodja za podatkovno analitiko, kot so Apache Spark, Hive, Pig itn. Kombinacij je toliko, koliko je orodij in ponujenih storitev na trgu, pri čemer načeloma omejitev ni, temveč so zgolj morebitni zadržki zaradi zapletenosti upravljanja takšne IT-arhitekture. Prednosti takšnih morebitnih kombinacij so vsekakor manjši stroški ter nezavezanost enemu ponudniku storitev. Prav tako pa bi lahko IT-arhitekti uporabili določene ponudnike oblačnih storitev zgolj za infrastrukturo kot storitev (IaaS), med tem ko bi sami nameščali in upravljali odprtokodne rešitve za del podatkovnega inženirstva ali za podatkovno analitiko.

Eden večjih izzivov pri načrtovanju arhitekture sodobnega cevovoda je predvsem izbira najbolj ustrezne rešitve za hitro in učinkovito shranjevanje podatkov ter rešitve za preprosto in morebitno kompleksno obdelavo in analizo le-teh. Zaradi tega se podjetja pogosto odločajo za izbiro oblačne storitve, ki predstavlja sistem OLTP (relacijski ali nerelacijski), saj se na ta način izognejo stroškom razširjanja in vzdrževanja obstoječih sistemov OLTP, lahko pa preprosto nadgradijo svoje obstoječe baze oz. najdejo kompatibilno oblačno rešitev kot nadgradnjo obstoječih baz (npr. Google Cloud SQL podpira implementacijo klasične baze MySQL). Na ravni podatkovnega jezera zadostuje hramba podatkov v obliki objektov ali datotek, kar veliko podjetij že uporablja zaradi shranjevanja vseh podatkov v sistemih. To pomeni, da je za podjetja dovolj, da so podatki v obliki objektov ali datotek ustrezno organizirani glede na potrebe prihodnjih analiz in njihove uporabe v naslednjem koraku cevovoda. Glede na to, da večina rešitev za podatkovna jezera v oblaku temelji na odprtokodnem HDFSu, je odločitev o uporabi rešitve za podatkovno jezero predvsem odvisna od možnosti samostojnega vzdrževanja rešitve. Podjetja z večjim proračunom se ponavadi odločajo za uporabo zanesljive oblačne rešitve, kot je Amazon S3 ali Google Object Storage, pri drugih komponentah cevovoda pa poskusijo biti čim bolj neodvisni od plačljivega ponudnika in uporabiti nadgrajeno različico komponente, ki jo že uporabljajo (npr. transakcijska podatkovna baza ali skladišče, ročno razviti proces ETL, brezplačna orodja za vizualizacijo ali podatkovno analitiko).

6 SKLEP

V članku smo predstavili teoretično ozadje načrtovanja sodobnih IT-arhitektur za upravljanje velepodatkov ter osnovne komponente takšnih sistemov. Področje upravljanja velepodatkov se tehnološko intenzivno razvija v zadnjem desetletju. Kot rezultat so nastale številne tehnologije različnih ponudnikov na trgu, kar predstavlja velik izziv pri izbiri tehnološkega sklada za podjetja, ki želijo izboljšati zmogljivost svojih obstoječih sistemov na ravni velepodatkov. Čez leta so se uporabljale različne rešitve za shranjevanje in obdelavo velikih količin podatkov za namene OLTP ali OLAP, kot so transakcije relacijske in nerelacijske podatkovne baze, podatkovna skladišča, podatkovna jezera in (po novem) podatkovna kolišča. Slednje se vključujejo kot komponente za shranjevanje podatkov v sodobnih podatkovnih cevovodih, ki zagotavljajo pravilne podatkovne toke, od podatkovnih virov do ciljne aplikacije ali orodij za podatkovno analitiko. Točnost in učinkovitost upravljanja podatkov v tistih cevovodih se zagotovita z ustreznim podatkovnim inženirstvom in IT-arhitekturo, ki

Tabela 1: **Pregled tehnoloških rešitev za implementacijo dvostopenjske arhitekture znotraj skladov Google, Amazon in Apache.**

	Sklad Google	**Sklad Amazon**	**Sklad Apache**
Podatkovno jezero	Google Cloud Storage	Amazon S3	Hadoop HDFS
Podatkovno skladišče (OLAP)	Google BigQuery	Amazon RedShift + RedShift Spectrum	- Apache Hive - Apache Doris
Podatkovna baza (OLTP)	- Google Cloud SQL ali Cloud Spanner (relacijska PB) - Google BigTable (nerelacijska PB) - Google Cloud MemoryStore (predpomnilnik)	Amazon Aurora	PostgreSQL + Citus
Podatkovna analitika in strojno učenje*	- Vertex AI in BigQuery ML - Apache Spark / Presto / Hive / Pig (vključuje uporabo rešitve Google DataProc)	- Amazon QuickSight - Amazon SageMaker	- Apache Hive - Apache Spark

vključuje eno ali več rešitev za zajem, shranjevanje, obdelavo ter dostop do podatkov. Že več let je najbolj priljubljena dvostopenjska arhitektura, saj hkrati zagotavlja hitrost poizvedb po transakcijskih podatkih, shranjenih v sistemih OLTP (oblačne ali on premise podatkovne baze), in učinkovitost kompleksnih poizvedb nad združenimi podatki, shranjenih v sistemih OLAP (podatkovna skladišča ali prilagojene širokostolpčne baze).

Kot je razvidno iz narejene analize tehnoloških skladov Google in Amazon, velike razlike med tema plačljivima ponudnikoma ni, saj sta arhitektura in način delovanja podatkovnega cevovoda v obeh primerih podobni in so uporabnikom na voljo močne in zelo razširljive rešitve. Prav tako temeljijo storitve, ki jih ponujajo na skoraj enakih osnovah ter se zgolj promovirajo z ločenimi blagovnimi znamkami in dodatnimi funkcionalnostmi. Z odprtokodnim skladom imajo Apache uporabniki večji nadzor nad posameznimi komponentami arhitekture, vendar to zahteva neprekinjeno nadzorovanje sistema zaradi pravočasnega ukrepanja za odpravljanje napak in skaliranje rešitev. Izbira tehnološkega sklada je torej odvisna od več dejavnikov, med katerimi sta najpomembnejši dostopnost človeških in finančnih virov v podjetju ter želena prilagodljivost in stopnja nadzora nad celotnim sistemom.

V tem članku smo se predvsem omejili na podatkovno inženirstvo in zagotavljanje IT-arhitekture za učinkovito uporabo in upravljanje velepodatkov, pri čemer pa smo izpustili določene izzive sodobnega podatkovnega inženirstva, kot so zahteve po realnočasovni obdelavi dogodkovnih podatkov itn. Prav tako smo analizirali in predstavili zgolj dva od treh velikih ponudnikov oblačnih storitev (tj. Google in Amazon). V prihodnosti bomo zajeli tudi analizo arhitektur Lambda in Kappa ter trenutno analizo razširili tudi na Microsoftov sklad in druge ponudnike, kot tudi predstaviti morebitne IT-arhitekture, ki kombinirajo uporabo komponent različnih ponudnikov. Prav tako se bomo osredotočili na analizo učinkovitosti in smotrnosti uporabe zgolj infrastrukture kot storitve (angl. Infrastructure-as-a-Service, IaaS) ter na osnovi tega snovanja IT-arhitekture, ki temelji na skladu Apache.

ZAHVALA

Rezulati so delno financirani s strani raziskovalnega programa št. P2-0057 Javne agencije za raziskovalno dejavnost Republike Slovenije iz državnega proračuna ter projektov ZeRoW (1001036388) in Data4Food2030 (101059473) financiranih iz okvirnega programa EU za raziskave in inovacije – Obzorje H2020 in Obzorje Evropa.

LITERATURA

[1] Memoona J Anwar, Asif Q Gill, Farookh K Hussain, and Muhammad Imran. Secure big data ecosystem architecture: challenges and solutions. EURASIP Journal on Wireless Communications and Networking, 2021(1):1–30, 2021.

[2] Michael Armbrust, Ali Ghodsi, Reynold Xin, and Matei Zaharia. Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In Proceedings of CIDR, 2021.

[3] Abhay Kumar Bhadani and Dhanya Jothimani. Big data: challenges, opportunities, and realities.

[4] Effective big data management and opportunities for implementation, pages 1–24, 2016.

[5] Citus Data. Citus dokumentacija. [Na spletu]. Dostopno: https://docs.citusdata.com/en/v11.1/get_started/what_is_citus.html. [Dostopano: 27-okt-2022], 2022.

[6] The Apache Software Foundation. Apache Sqoop dokumentacija. [Na spletu]. Dostopno: https://sqoop.apache.org/. [Dostopano: 27-okt-2022], 2019.

[7] The Apache Software Foundation. Apache Doris dokumentacija. [Na spletu]. Dostopno: https://doris.apache.org/. [Dostopano: 27-okt-2022], 2022.
[8] The Apache Software Foundation. Apache Hive dokumentacija. [Na spletu]. Dostopno: https://hive.apache.org/. [Dostopano: 27-okt-2022], 2022.
[9] The Apache Software Foundation. Hadoop HDFS Architecture Guide. [Na spletu]. Dostopno: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html. [Dostopano: 27-okt-2022], 2022.
[10] Paramita Ghosh. Data architecture challenges. [Na spletu]. Dostopno: https://www.dataversity.net/data- architecture--challenges. [Dostopano: 4-okt-2022], junij 2022.
[11] Josh Howarth. 30+ incredible big data statistics (2022). [Na spletu]. Dostopno: https://explodingtopics.com/blog/big-data-stats. [Dostopano: 3-okt-2022], avgust 2022.
[12] Oliver Hummel, Holger Eichelberger, Andreas Giloj, Dominik Werle, and Klaus Schmid. A collection of software engineering challenges for big data system development. In 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pages 362–369. IEEE, 2018.
[13] Amazon Web Services Inc. AWS Serverless Data Analytics Pipeline. [Na spletu]. Dostopno: https://docs.aws.amazon.com/pdfs/whitepapers/latest/aws-serverless-data-analytics--pipeline/aws- serverless-data-analytics-pipeline.pdflogical--architecture-of-modern-data-lake-centric-analytics- platforms. [Dostopano: 27-okt-2022], april.
[14] Amazon Web Services Inc. Amazon Aurora dokumentacija. [Na spletu]. Dostopno: https://aws.amazon.com/rds/aurora/. [Dostopano: 27-okt-2022], 2022.
[15] Amazon Web Services Inc. Amazon AWS Glue dokumentacija. [Na spletu]. Dostopno: https://aws.amazon.com/glue/. [Dostopano: 27-okt-2022], 2022.
[16] Amazon Web Services Inc. Amazon AWS S3 dokumentacija. [Na spletu]. Dostopno: https://aws.amazon.com/s3/. [Dostopano: 27-okt-2022], 2022.
[17] Amazon Web Services Inc. Amazon RedShift dokumentacija. [Na spletu]. Dostopno: https://aws.amazon.com/redshift/. [Dostopano: 27-okt-2022], 2022.
[18] Amazon Web Services Inc. AWS Glue Components. [Na spletu]. Dostopno: https://docs.aws.amazon.com/glue/latest/dg/components-overview.html. [Dostopano: 27-okt-2022], 2022.
[19] Amazon Web Services Inc. Central storage: Amazon S3 as the data lake storage platform. [Na spletu]. Dostopno: https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/amazon-s3-data-lakestorage-platform.html. [Dostopano: 27-okt-2022], 2022.
[20] Amazon Web Services Inc. Getting started with Amazon Redshift Spectrum. [Na spletu]. Dostopno: https://docs.aws.amazon.com/redshift/latest/dg/c-getting-started-using--spectrum.html. [Dostopano: 27-okt-2022], 2022.
[21] Google Inc. Google Cloud BigQuery dokumentacija. [Na spletu]. Dostopno: https://cloud.google.com/bigquery. [Dostopano: 27-okt-2022], 2022.
[22] Google Inc. Google Cloud BigTable dokumentacija. [Na spletu]. Dostopno: https://cloud.google.com/bigtable. [Dostopano: 27-okt-2022], 2022.
[23] Google Inc. Google Cloud DataProc dokumentacija. [Na spletu]. Dostopno: https://cloud.google.com/dataproc. [Dostopano: 27-okt-2022], 2022.
[24] Google Inc. Google Cloud MemoryStore dokumentacija. [Na spletu]. Dostopno: https://cloud.google.com/memorystore. [Dostopano: 27-okt-2022], 2022.
[25] Google Inc. Google Cloud Spanner dokumentacija. [Na spletu]. Dostopno: https://cloud.google.com/spanner. [Dostopano: 27-okt-2022], 2022.
[26] Google Inc. Google Cloud SQL dokumentacija. [Na spletu]. Dostopno: https://cloud.google.com/sql. [Dostopano: 27-okt-2022], 2022.
[27] Google Inc. Google Cloud Storage dokumentacija. [Na spletu]. Dostopno: https://cloud.google.com/storage. [Dostopano: 27-okt-2022], 2022.
[28] Godson Koffi Kalipe and Rajat Kumar Behera. Big data architectures: A detailed and application oriented review. International Journal of Innovative Technology and Exploring Engineering, 8:2182–2190, 2019.
[29] Avita Katal, Mohammad Wazid, and Rayan H Goudar. Big data: issues, challenges, tools and good practices. In 2013 Sixth international conference on contemporary computing (IC3), pages 404–409. IEEE, 2013.
[30] Jimmy Lin. The lambda and the kappa. IEEE Internet Computing, 21(05):60–66, 2017.
[31] Gaurav Mishra. Setting up a Data Lake architecture with AWS. [Na spletu]. Dostopno: https://www.srijan.net/resources/blog/setting-up-a-data-lake-architecture-with-aws. [Dostopano: 27-okt-2022], avgust 2019.
[32] Pointer, Ian. What is Apache Spark? The big data platform that crushed Hadoop. [Na spletu]. Dostopno: https://www.infoworld.com/article/3236869/what-is-apache-spark-the--big-data-platform-that- crushed-hadoop.html. [Dostopano: 27-okt-2022], marec 2020.
[33] Zhi-Hua Zhou, Nitesh V Chawla, Yaochu Jin, and Graham J Williams. Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]. IEEE Computational intelligence magazine, 9(4):62–74, 2014.

▪

Martina Šestak je asistentka z doktoratom in raziskovalka na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Trenutno se raziskovalno ukvarja s sodobnimi arhitekturami za upravljanje velepodatkov, podatkovnimi bazami grafov in podatkovnimi prostori. Raziskovalno in aplikativno sodeluje tudi na več projektih, ki se odvijajo v okviru Inštituta za informatiko.

▪

Muhamed Turkanović je visokošolski učitelj, izredni profesor, na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Je vodja raziskovalne skupine Blockchain Lab:UM Inštituta za informatiko, namestnik predstojnika Inštituta za informatiko, vodja slovenskega EDIH-a DIGI-SI, vodja Digitalnega inovacijskega stičišča Univerze v Mariboru, vodja projektov H2020, Horizont Evropa, Interreg Alpine Space ter ARRS CRP. Njegovi trenutni raziskovalni interesi vključujejo področja tehnologij veriženja blokov, podatkovnih tehnologij ter digitalnih identitet.