Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Alja Kunovar

Pregled in primerjava nerelacijskih podatkovnih baz kot storitev

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: doc. dr. Dejan Lavbič

Ljubljana 2013

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00088/2013
Datum: 15.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ALJA KUNOVAR**

Naslov: **PREGLED IN PRIMERJAVA NERELACIJSKIH PODATKOVNIH BAZ KOT STORITEV**

REVIEW OF NOSQL DATABASES AS A SERVICE

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

NoSQL podatkovne baze postajajo vse bolj uporabljana alternativa relacijskim podatkovnim bazam tudi v poslovnem okolju. Kljub temu, da gre za nezrelo tehnologijo, obstajajo številne storitve v oblaku, ki omogočajo gostovanje teh omenjenih podatkovnih baz. Med posameznimi rešitvami obstajajo številne razlike, ki se predvsem izražajo v funkcionalnostih in obračunavanju storitve. V okviru diplomske naloge preglejte obstoječe ponudnike in za vsako skupino NoSQL podatkovnih baz izberite predstavnika, ki ga podrobno predstavite in analizirate. Pri primerjavi identificirajte kriterije ter pripravite odločitveni model, ki temelji na večkriterijskem odločanju. Posamezne predstavnike ovrednotite in rezultate predstavite v obliki radarskega grafa.

Mentor:

doc. dr. Dejan Lavbič

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic

Izjava o avtorstvu diplomskega dela

Spodaj podpisana Alja Kunovar, z vpisno številko **63100218**, sem avtorica diplomskega dela z naslovom:

Pregled in primerjava nerelacijskih podatkovnih baz kot storitev

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Dejana Lavbiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 1. septembra 2013 Podpis avtorja:

Zahvala

Zahvaljujem se mentorju, doc. dr. Dejanu Lavbiču, za pomoč in koristne nasvete pri izdelavi diplomskega dela.

Posebna zahvala gre tudi g. Jamesu Andersonu, ki mi je omogočil uporabo podatkovne baze kot storitev Dydra, ki je trenutno še v zaprti beta različici.

Kazalo

Seznam uporabljenih kratic in simbolov

ACID (Atomicity, Consistency, Isolation, Durability) atomarnost, konsistentnost, izolacija in trajnost so lastnosti transakcij v relacijski podatkovni bazi.

API (Application Programming Interface) je aplikacijski programski vmesnik.

AWS (Amazon Web Services) so Amazonove spletne storitve.

BASE (Basical Availability, Soft state, Eventual Consistency) osnovna razpoložljivost, mehko stanje in zakasnjena konsistentnost so lastnosti modela, ki ga uporabljajo nerelacijske podatkovne baze.

BLOB (Binary Large Object) je veliki binarni objekt.

BSON (Binary JSON) je binarni format za opisovanje podatkov, ki ga uporablja MongoDB.

CAP (Consistency, Availability, Partition Tolerance) konsistentnost, razpoložljivost in particijska toleranca.

DBaaS (Database as a Service) je podatkovna baza kot storitev.

HTTP (HyperText Transfer Protocol) je protokol, ki skrbi za ustrezen prenos informacij preko spleta.

IaaS (Infrastructure as a Service) je infrastruktura kot storitev.

IOQ (Input Output Queue) je vhodna/izhodna vrsta za določanje prioritet zahtev.

JSON (JavaScript Object Notation) je format za opisovanje podatkov, ki temelji na jeziku JavaScript.

NaaS (Network as a Service) je omrežje kot storitev.

NoSQL (Not Only SQL) je oznaka za novo generacijo nerelacijskih podatkovnih baz.

PaaS (Platform as a Service) je platforma kot storitev.

RDF (Resource Description Framework) je osnovni format za zapis podatkov na semantičnem spletu.

REST (Representational State Transfer) je množica arhitekturnih principov za razvoj spletnih storitev.

SaaS (Software as a Service) je platforma kot storitev.

SDK (Software Development Kit) je paket za razvoj programske opreme.

SPARQL (SPARQL Protocol and RDF Query Language) je poizvedovalni jezik za format RDF.

SQL (Structured Query Language) je poizvedovalni jezik v relacijskih podatkovnih bazah.

URL (Uniform Resource Locator) je enolični krajevnik vira, ki določa spletno stran v svetovnem spletu.

UTF-8 (UCS Transformation Format-8bit) je eden izmed načinov kodiranja nabora znakov Unicode.

W3C (World Wide Web Consortium) je mednarodni inštitut, ki razvija standarde za splet.

XML (Extensible Markup Language) je razširljiv označevalni jezik, ki omogoča format za opisovanje strukturiranih podatkov.

Povzetek

V zadnjih nekaj letih so v porastu računalniške storitve v oblaku, med katere sodijo tudi podatkovne baze kot storitev. Relacijske podatkovne baze pri shranjevanju velikih količin podatkov ne dosegajo želenih rezultatov, zato se vedno bolj uveljavljajo nerelacijske podatkovne baze. Ker ljudje dandanes svojo osredotočenost preusmerjajo na posel in morebitne nadgradnje aplikacij, želijo načrtovanje infrastrukture podatkovne baze in upravljanje s podatki prepustiti drugim. Zaradi navedenega bomo v tem diplomskem delu najprej predstavili pet ponudnikov nerelacijskih podatkovnih baz kot storitev, ki se razlikujejo glede na kategorije podatkovne baze. S tem diplomskim delom želimo uporabnikom olajšati izbiro ponudnika nerelacijskih podatkovnih baz kot storitev, upoštevaje uporabnikove zahteve. Spoznali bomo, da ponudniki svoje storitve ponujajo v naročniških paketih, ki zajemajo različne funkcionalnosti, obračunavanje storitev na več različnih načinov in različne stopnje podpore uporabnikom.

Ključne besede: nerelacijske podatkovne baze, NoSQL, računalništvo v oblaku, podatkovna baza kot storitev

Abstract

In the last few years there has been an increase of cloud services, among which databases as a service are also included. Relational databases do not meet desired results for storing large quantities of data, therefore the importance of NoSQL databases is growing. Because people nowadays are redirecting their focus on business and application upgrades, they want to leave database infrastructure planning and data management to others. Having said that, we present five providers of NoSQL databases as a service, which vary depending on the category of the database. In this thesis, we want to make easier the selection of providers of NoSQL databases as a service, considering user requirements. We will show that providers offer their services in subscription plans that include different features, use different charging methods and offer different levels of user support.

Keywords: nonrelational databases, NoSQL, cloud computing, database as a service

Poglavje 1

Uvod

V zadnjih desetletjih se informacijske tehnologije razvijajo z bliskovito hitrostjo, zato trenutno ni mogoče predvideti, da se bo razvoj v naslednjih letih upočasnil. Eden izmed zadnjih velikih prelomov v informacijskih tehnologijah, ki je dodobra spremenil poslovanje podjetij, je pojav računalništva v oblaku. Oblikovalo se je več storitvenih modelov računalništva v oblaku, med njimi tudi podatkovne baze kot storitve. Ker je v zadnjem času povprašanjevanje po nerelacijskih podatkovnih bazah v primerjavi z relacijskimi podatkovnimi bazami v porastu, smo se odločili, da združimo obe tematiki in v tem diplomskem delu predstavimo nerelacijske podatkovne baze kot storitve.

Ljudje se vedno bolj pogosto odločajo, da bodo svoje podatke, namesto na lastnih strežnikih, shranjevali na tujih strežnikih, ki so administrirani s strani strokovnjakov. Zato imajo sedaj možnost uporabiti tujo storitev, ki nadomešča čas, denar in delovno silo, ki so bili doslej porabljeni za vzpostavljanje lastne podatkovne infrastrukture. Tako se lahko sedaj uporabniki namesto na vzpostavljanje in konfiguracijo strežnikov, porazdeljevanje podatkov in vzdrževanje podatkovne baze, osredotočijo na nadaljni razvoj in uporabo aplikacije, kar je ključnejšega pomena. Uporaba podatkovne baze kot storitve, v primerjavi z lastno podatkovno infrastrukturo, močno zniža začetni kapital, ki je potreben za razvoj aplikacij. Navedeno pripomore k

lažjemu začetku vedno pogostejših startup podjetij, saj jim prav začetni kapital predstavlja največjo oviro.

Trenutno največ ljudi uporablja podatkovne baze kot storitve za eno izmed naslednjih dejavnosti: razvoj in testiranje, spletne aplikacije, dodajanje kapacitet, varnostne kopije in analitične aplikacije [1].

Na voljo je kar nekaj različnih ponudnikov podatkovnih baz kot storitev, ki se povečini razlikujejo v kategoriji podatkovne baze, ki jo uporabljajo, ponujenih funkcionalnostih, ceni ter kakovosti storitve.

Cilj tega diplomskega dela je predstaviti nekaj ponudnikov nerelacijskih podatkovnih baz kot storitev in opredeliti kriterije, ki bodo služili za pomoč pri izbiri najustreznejšega ponudnika.

Diplomsko delo je razdeljeno na pet poglavij. V uvodnem poglavju je predstavljena tema in namen diplomske naloge, v drugem poglavju pa opis pojma računalništvo v oblaku, kjer bomo navedli pogoje, ki jih mora izpolnjevati podatkovna baza kot storitev, ter razloge za uporabo le-te. Nato sledi tretje poglavje o nerelacijskih podatkovnih bazah, kjer bomo sprva predstavili teoretične koncepte nerelacijskih podatkovnih baz, in sicer pomen modelov ACID in BASE ter teorem CAP. V istem poglavju je predstavljen tudi pregled in opis ponudnikov nerelacijskih podatkovnih baz kot storitev, glede na kategorijo podatkovne baze. Četrto poglavje je namenjeno primerjavi ponudnikov, kjer bomo opisali kriterije primerjave in sklepe do katerih smo prišli. V petem poglavju pa je predstaljen postopek izgradnje odločitvenega modela ter ugotovitve, ki so predstavljene na grafikonih. V zadnjem poglavju so zajete sklepne ugotovitve diplomskega dela.

Poglavje 2

Računalništvo v oblaku

Definicija računalništva v oblaku, povzeta po pojasnilih svetovne raziskovalne družbe Gartner, pravi, da je računalništvo v oblaku slog računalništva, kjer so visoko skalabilne in prilagodljive zmogljivosti informacijskih tehnologij dostopne zunanjim odjemalcem kot storitev, z uporabo internetnih tehnologij [2].

Računalništvo v oblaku prinaša prednosti na mnogih področjih, pri kupcih od zmanjševanja vrednosti začetnih naložb v informacijsko in komunikacijsko tehnologijo, nižanja stroškov strojne in programske opreme, do uvajanja in vzdrževanja. Prinaša tudi številne koristi za ponudnike, ki lahko sedaj razvijajo in vzdržujejo aplikacije na enotni platformi.

Przemek Sienkiewicz, vodja enote za poslovne storitve za vzhodno Evropo in Rusijo podjetja Google, je kot tuji gost na 5. posvetu dolenjskih in belokranjskih informatikov zatrdil, da so ravno orodja računalništva v oblaku ključne rešitve, kako podpirati sodoben in uspešen način poslovanja, saj omogočijo delovno mesto brez omejitev in podpirajo sodelovanje med različnimi deli podjetij, kar je lahko celo ključen element uspeha [3].

Računalništvo v oblaku sestavlja več storitvenih modelov, in sicer:

- IaaS (infrastruktura kot storitev)
- PaaS (platforma kot storitev)

- Saas (programska oprema kot storitev)
- NaaS (slo. omrežje kot storitev)

Znotraj storitvenega modela programska oprema kot storitev (SaaS) lahko opredelimo tudi podmodel, ki se imenuje podatkovna baza kot storitev (ang. *Database as a Service*). Podatkovna baza kot storitev nudi odjemalcu programsko opremo za podatkovne baze in fizično shranjevanje podatkov v baze. Pod pojmom podatkovna baza kot storitev lahko opredelimo vse storitve, ki izpolnjujejo naslednje pogoje:

- Storitev mora biti na voljo stranki na zahtevo (ang. *on-demand*), brez kakršnekoli zahteve za namestitev in konfiguracijo strojne ali programske opreme.
- Storitev se zaračuna kupcu, glede na količino porabe, ki se izračuna na podlagi različnih parametrov, brez kakršnekoli zahteve po dolgoročni pogodbi ali vnaprejšnjem plačilu. Tako plačamo le za tiste storitve, ki jih dejansko uporabimo.
- Ponudnik je odgovoren za upravljanje storitve, zato mora on skrbeti, da bo strankina podatkovna baza vzdrževana in ažurna [4].

Navedeni pogoji morajo biti izpolnjeni tudi pri drugih vejah računalništva v oblaku. Poleg zgoraj zahtevanih lastnosti je od arhitekture DBaaS pričakovano tudi, da podpira elastičnost storitve, varno več-najemništvo (ang. *multi-tenancy*), avtomatsko upravljanje z viri in integrirano skalabilnost. Našteti atributi niso specifični le za podatkovno bazo kot storitev, temveč so bistvenega pomena tudi za tradicionalne arhitekture podatkovnih baz. DBaaS pa se od tradicionalne arhitekture podatkovnih baz razlikuje v svoji usmerjenosti v storitve in osredotočenju na možnost, da stranka sama dodeljuje, upravlja in zaključuje storitve preko spletnega ali programskega vmesnika (self-service).

Najpogostejši razlogi za odločitev strank za uporabo podatkovne baze kot storitve so sledeči:

- **Nižji stroški** - Vodilni v podjetjih se zavedajo potrebe po točnih in ažurnih podatkih pri sprejemanju pomembnih odločitev, ki vplivajo na poslovanje podjetja. Da bi zagotovili, da so ob potrebnem času na voljo prave informacije, zaposleni na projektih namenijo velik del svojega časa, virov in denarja za ustvarjanje informacijskih silosov z različnimi konfiguracijami. DBaaS pa daje podjetjem možnost za standardizacijo in optimizacijo na platformi, ki odpravlja vsakokratno potrebo po upravljanju in vzdrževanju strojne in programske opreme za vsako okolje posebej. Arhitektura DBaaS je namreč izdelana tako, da omogoča elastičnost storitve in akumulacijo virov.

- **Izboljšana raven storitve** - Od izboljšane ravni storitve imajo koristi ne le stranke, temveč tudi ponudniki. Ponudniki z razvijanjem vnaprej določene ponudbe z razvojem standardiziranih postopkov in skupnih mehanizmov za podporo zmanjšajo raznolikost konfiguracij in programske opreme. Navedeno podpira njihove poslovne cilje, ki stremijo k agilnosti, učinkovitosti in izboljšani kakovosti storitve, stranke pa točno vedo kakšne zmogljivosti in raven storitve lahko pričakujejo od ponudnika. Ponudnik omogoča stranki tudi celovit pregled o uporabi storitev in stroških, ki so ob tem nastali, kar pripomore k boljši ozaveščenosti stranke o porabljenem denarju.

- **Izboljšan dostop do informacij in racionalizacija** - Uveljavljeno je napačno prepričanje, da so zahteve po podatkih tako raznolike, da vsaka poslovna dejavnost zahteva svojo podatkovno bazo. DBaaS namreč omogoča, da imajo lastniki podatkov možnost nadzora nad tem, kdaj so podatki v skupni rabi, kar odpravlja potrebo po odvečnih podatkovnih bazah in omogoča večjo varnost [5].

Poglavje 3

Nerelacijske podatkovne baze

Vrsto let so bile relacijske podatkovne baze odgovor na številne probleme na katere smo naleteli pri izbiri podatkovne baze. Čeprav se v veliki meri uporabljajo še danes, pa veliko ljudi želi zamenjati relacijske sisteme za upravljanje podatkovnih baz za podatkovne baze z naslednjimi lastnostmi:

- zmožnost horizontalnega skaliranja (ang. *horizontal scalability*) čez več strežnikov
- zmožnost replikacije in particioniranja podatkov čez več strežnikov
- šibkejši model nadzora nad sočasnim izvajanjem transakcij kot pri ACID transakcijah
- zmožnost dostopa s preprostim vmesnikom API
- zmožnost dinamičnega dodajanja novih atributov trenutnim zapisom [6]

Podatkovne baze s takšnimi lastnostmi imenujemo nerelacijske podatkovne baze, leta 2009 pa se je zanje začel uporabljati izraz "NoSQL", ki pomeni "Not Only SQL"in hrati ponazarja, da te vrste podatkovnih baz ne podpirajo poizvedovalnega jezika SQL in niso relacijske. Izraz NoSQL je prvi uporabil Carlo Strozzi, leta 1998, za poimenovanje svoje odprtokodne relacijske podatkovne baze, ki ni podpirala standardnega SQL vmesnika. Strozzi

je tudi predlagal, da bi se NoSQL gibanje moralo preimenovati v primernejše ime "NoREL", zaradi oddaljevanja od tradicionalnega relacijskega modela [7].

3.1 Osnovni koncepti nerelacijskih podatkovnih baz

3.1.1 Modela ACID in BASE

Za razliko od relacijskih baz, nerelacijske podatkovne baze svojim transakcijam ne zagotavljajo lastnosti ACID. ACID je kratica za atomarnost (***A****tomicity*), konsistentnost (***C****onsistency*), izolacijo (***I****solation*) in trajnost (***D****urability*). Gre za skupek pravil, ki zagotavljajo, da transakcije v podatkovni bazi vodijo iz enega veljavnega stanja v drugo veljavno stanje. Namesto modela ACID, ki pri nekaterih situacijah ovira delovanje podatkovne baze, se pri NoSQL bazah uporablja model BASE. BASE je kratica za osnovno razpoložljivost (***B****asic* ***A****vailability*), mehko stanje (***S****oft state*) in zakasnjeno konsistentnost (***E****ventual Consistency*). Ta model podpira prilagodljivost in druge pristope, ki se uporabljajo za delo z nestrukturiranimi podatki, ponujene s strani NoSQL podatkovnih baz.

Model BASE sestoji iz treh principov:

- **Osnovna razpoložljivost** - NoSQL podatkovne baze se osredotočajo na razpoložljivost podatkov, tudi v primeru prisotnosti večih napak. Razpoložljivost napak se dosega z uporabo pristopa močno porazdeljenih (ang. *highly distributed*) podatkovnih baz. Namesto, da bi vzdrževali eno veliko shrambo podatkov in skrbeli, da ne pride do napak, raje uporabimo NoSQL bazo, ki razdeli podatke čez več shramb podatkov z visoko stopnjo replikacije. Če v tem primeru pride do napake dostopa do podatka, to ne vodi v popoln izpad delovanja podatkovne baze.

- **Mehko stanje** - Model BASE popolnoma opusti zahtevo po konsistentnosti, ki je eden izmed principov v modelu ACID. Stanje sistema se tako zaradi posodobitev čez čas lahko spremeni tudi brez novih vnosov. Posledica tega je, da so kopije podatkov v nekem trenutku lahko nekonsistentne.

- **Zakasnjena konsistentnost** - Edina zahteva NoSQL podatkovnih baz po konsistentnosti je ta, da bodo v nekem poljubnem času v prihodnosti podatki konsistentni, vendar ni nobenega jamstva, kdaj bo stanje zagotovljeno [8].

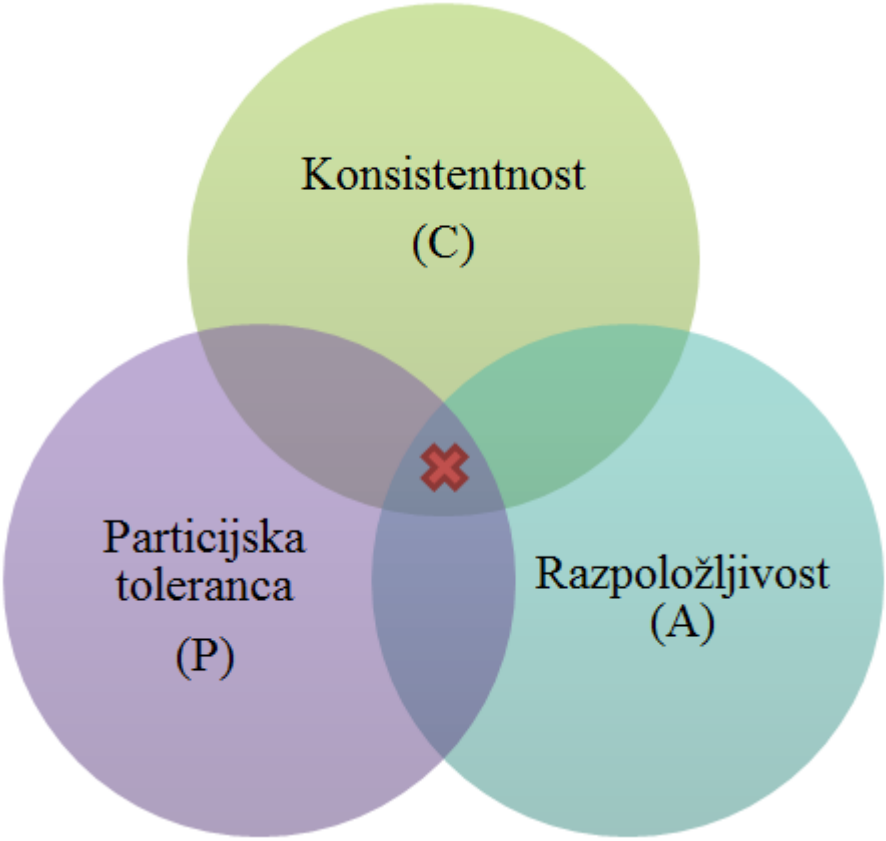
3.1.2 CAP teorem

Teorem CAP je poznan tudi pod imenom Brewerjev teorem, po profesorju na Berkeleyu, Univerzi v Kaliforniji, ki je, leta 2000, na Simpoziju principov porazdeljenega računalništva (ang. *Symposium on Principles of Distributed Computing*) [9] prvi predstavil svoje domneve glede porazdeljenih sistemov. Dve leti kasneje sta Seth Gilbert in profesorica Nancy Lynch iz MIT-ja v akademskem članku z naslovom "Brewerjeve domneve in potrebe po konsistentnih, razpoložljivih in particijsko tolerantnih spletnih storitvah"(ang. *Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services* [10]) potrdila njegove domneve in njegov teorem je postal sprejet s strani gibanja NoSQL.

CAP je kratica, ki predstavlja tri lastnosti porazdeljenih sistemov:

- **Konsistentnost** (ang. ***C****onsistency*) - Konsistentnost pri teoremu CAP nima enakega pomena kot pri modelu ACID. Pri modelu ACID konsistentnost pomeni, da se podatki, ki ne ustrezajo vnaprej določenim omejitvam, ne ohranijo. Pri teoremu CAP pa se povezuje z atomarnostjo in izolacijo branj in pisanj. Konsistentnost pomeni, da sočasne operacije branj in pisanj vidijo iste veljavne in konsistentne podatke.

- **Razpoložljivost** (ang. ***A****vailability*) - Razpoložljivost pomeni, da je sistem na voljo za delo, ko je to potrebno.



Slika 3.1: Diagram sprejemanja kompromisov pri teoremu CAP.

- **Particijska toleranca** (ang. ***P****artition tolerance*) - Particijska toleranca pomeni sposobnost sistema, da nemoteno deluje, tudi v primeru izpada nekaterih vozlišč v gruči [11].

Teorem CAP pravi, da je v porazdeljenih sistemih nemogoče doseči vse tri lastnosti istočasno, zagotavljamo lahko le dve izmed treh lastnosti. Izbira dveh lastnosti pa je prepuščena uporabniku glede na njegov problem.

3.2 Kategorije podatkovnih baz

3.2.1 Ključ-vrednost podatkovne baze

Ključ-vrednost podatkovna baza je najenostavnejši model nerelacijske baze, na katerem temeljijo ostale nerelacijske podatkovne baze. Kot že ime pove, ključ-vrednost podatkovna baza vsebuje množico parov ključ-vrednost. Vsak ključ je unikaten in se potlej preslika v pripadajočo vrednost, podobno kot pri seznamih v nekaterih programskih jezikih. Dolžina ključa je ponavadi omejena na določeno število bajtov, medtem ko je manj omejitev pri vrednosti. Vrednost je poljubnega podatkovnega tipa, saj jih podatkovna baza jih shranjuje kot objekte BLOB in se ne zaveda njihove vsebine. Zato pa se mora vsebine zavedati aplikacija. Podatkovne baze tipa ključ-vrednost uporabljajo primarni ključ, zato so visoko zmogljive in skalabilne. Moderne ključ-vrednost podatkovne baze dajejo prednost visoki skalabilnostji pred konsistentnostjo, zato so pri mnogih opuščena ad-hoc poizvedovanja in analitične funkcije. Nekatere vrste ključ-vrednost podatkovnih baz omogočajo, da v njih shranjujemo zapletenejše podatkovne tipe, kar bomo spoznali v nadaljevanju [12, 13].
Predstavniki podatkovnih baz so: Riak [14], Redis [15], Berkeley DB [16], HamsterDB [17], DynamoDB [18], Project Voldemort [19].
Storitve pa nudijo: Amazon DynamoDB (DynamoDB) [18], Redis To Go (Redis) [20].

V nadaljevanju bomo predstavili ponudnika Amazon DynamoDB, saj je najbolj razširjena podatkovna baza kot storitev.

3.2.1.1 Amazon DynamoDB - DynamoDB

Amazon DynamoDB je le ena izmed mnogih spletnih storitev, ki jih nudi Amazon pod oznako Amazon Web Services (v nadaljevanju AWS). Temelji na principih podatkovne baze Amazon Dynamo in Amazon SimpleDB ter združuje dobre lastnosti obeh. Zaradi vedno večjega obsega podatkov, so razvili podatkovno bazo, ki nudi visoko razpoložljivost, skalabilnost in dobre

sposobnosti delovanja. Zaradi svojega poslovnega modela ima vsak izpad delovanja velik negativni finančni učinek, zato so zahteve glede odpora na napake in zmogljivosti zelo visoke [21].

Podatkovni model Podatkovni model Amazon DynamoDB vključuje tabele, elemente in atribute. Podatkovna baza je zbirka tabel, tabele so zbirke elementov in vsak element je zbirka atributov. Baza nima vnaprej predpisane sheme, potrebno je le določiti primarni ključ. Tabela ima lahko poljubno število atributov, omejitve pa so pri velikosti elementov, saj je vsak lahko velik do 64KB (velikost elementa je vsota dolžin imen atributov in njegovih vrednosti). Vsak atribut v elementu je tipa ključ-vrednost. Atribut ima lahko tudi več vrednosti, vendar morajo biti te različne. Primer elementa:

```
{
     id = 101
     NaslovKnjige = "Alamut"
     ISBN = "9781556436819"
     Avtor = "Vladimir Bartol"
     Vezava = [ "trda", "mehka" ]
}
```

DynamoDB loči dva tipa primarnega ključa in sicer zgoščevalni tip, ki je sestavljen iz enega atributa, ter zgoščevalni in razponski tip, ki je sestavljen iz dveh atributov. Omogoča tudi uporabo sekundarnih indeksov. Podpira poizvedbe tipa PUT/GET, ki uporabljajo ključ, definiran s strani uporabnika [22].

Značilnosti Amazon DynamoDB ima sposobnost horizontalnega skaliranja, ki omogoča nemoteno porazdelitev tabel čez več strežnikov. Nudi zelo napredno konzolo za upravljanje in ima podporo za API.

Število predvidenih operacij določimo ob kreiranju tabele. Vrednost lahko v nadaljevanju spreminjamo. Od števila operacij, čemur pravimo tudi kapaciteta pretoka, je odvisna cena storitve. Glede na vneseno vrednost, Amazon

DynamoDB samodejno alocira namenske vire in porazdeli podatke. Zagotavlja hitro delovanje, največja zakasnitvena latenca naj ne bi presegala 10 milisekund.

Pri operacijah branja lahko izberemo kakšno vrsto konsistentnosti želimo. Privzeta možnost je zakasnjena konsistentnost branja, ki zagotavlja nizko zakasnitev in visok pretok. Vendar te podatki niso vedno ažurni, saj ne upoštevajo ravnokar končanih pisanj. Zato lahko tam, kjer je potrebno, izberemo konsistentno branje, ki vedno vrne rezultat, ki je upošteval vsa pisanja.

Arhitektura Amazonova infrastruktura sestoji iz več deset tisoč strežnikov, ki se nahajajo v številnih podatkovnih centrih.

CAP teorem Amazon DynamoDB zagotavlja razpoložljivost in particijsko toleranco. To zagotavlja s sinhronsko replikacijo čez tri podatkovne centre. Vsa vozlišča so enakovredna in opravljajo enake naloge.

3.2.2 Dokumentno usmerjene podatkovne baze

Dokumentno usmerjene podatkovne baze so v osnovi vrste ključ-vrednost, ki v polje vrednost shranjujejo zapletenejše podatkovne strukture, to je dokumente. Razlog za nastanek dokumentno usmerjenih podatkovnih baz je pogosta uporaba spletnih storitev, ki za svoje delovanje uporabljajo dokumente XML, JSON in BSON. Zato je namesto, da se omenjeni dokumenti preslikajo v relacijo, ki jo predstavljajo vrstice in stolpci, bolj primerno shraniti dokument kot tak, ki omogoča shranjevanje vrednosti v obliki seznamov, skalarnih vrednosti in tudi ugnezdenih dokumentov. Za razliko od podatkovnih baz vrste ključ-vrednost, se dokumentno usmerjene podatkovne baze zavedajo podatkovne strukture v polju vrednost, kar je vzrok za boljšo prenosljivost, dostopnost ter podporo sekundarnemu indeksiranju in več tipom dokumentov oz. objektov na bazo [23, 13]. Dokumenti lahko vsebujejo tudi priponke v tradicionalnem smislu dokumentov (datoteke PDF, datoteke Mi-

crosoft Word, članke, itd.), vendar je lahko dokument katerekoli vrste objekt. Nekatere baze podpirajo tudi verziranje dokumentov, kjer se stare verzije ohranijo in oštevilčijo. Pomemben vidik dokumentno usmerjenih podatkovnih baz je, da se lahko naslavljajo z edinstvenim naslovom URL. Zaradi dobre URL in HTTP naravnanosti podpirajo arhitekturo REST. Tako lahko za poizvedovanje uporabljamo tudi vmesnike API [6].

Predstavniki podatkovnih baz so: MongoDB [24], CouchDB [25], RavenDB [26], Terrastore [27].
Storitve pa nudijo: Cloudant (CouchDB) [28], MongoLab (MongoDB) [29], MongoHQ (MongoDB) [30], MongoDirector (MongoDB) [31], ObjectRocket (MongoDB) [32], CloudBird (RavenDB) [33], RavenHQ (RavenDB) [34].
Predstavili bomo dva predstavnika dokumentno usmerjenih podatkovnih baz kot storitev, ki temeljita na najpogosteje uporabljenih dokumentno usmerjenih podatkovnih bazah, in sicer Cloudant, ki temelji na CouchDB, ter MongoLab, ki temelji na MongoDB. Izbira med ponudniki podatkovnih baz kot storitev, ki temeljijo na MongoDB je kar velika, odločili smo se za enega izmed brezplačnih ponudnikov.

3.2.2.1 Cloudant - CouchDB

Podatkovno bazo kot storitev Cloudant so leta 2008 ustanovili trije fiziki iz MIT-ja, ki so delali z več petabajtov podatkov velikega hadronskega trkalnika. Ker niso bili zadovoljni s trenutnimi orodji za upravljanje s tako velikimi količinami podatkov, so se odločili, da si sami izdelajo primerno orodje. Cloudant ponuja podatkovno bazo kot storitev, ki temelji na podatkovni bazi Apache CouchDB. Izbrali so jo, ker je odlična pri hkratnih dostopih, enostavna za uporabo zaradi API-ja za spletne storitve RESTful in podpira zmožnost replikacije in sinhronizacije različnih kopij podatkovne baze.

Podatkovni model Narejena je za shranjevanje pol-struktiranih podatkov, kot na primer naslednja dokumenta JSON.
Prvi primer:

```
{
"_id":"636a47478909452c50a461b9af1d4e49",
"_rev":"1-110c9616fa3b58d2b9499a56f63a98e5",
"Ime":"Janez Novak",
"Dejavnosti":[
     {"Naziv":"Smucanje","Letni cas":"zima"},
     {"Naziv":"Plavanje","Letni cas":"poletje"}
      ]
}
```

Drugi primer:

```
{
"_id":"636a47478909452c50a461b9af1d4e49",
"_rev":"2-fa290574c2fd9748eaf044c4404d2afc",
"Ime":"Janez Novak",
"Dejavnosti":[
     {"Naziv":"Smucanje","Letni cas":"zima"},
     {"Naziv":"Kolesarjenje","Letni cas": ["pomlad", "jesen"]}
      ]
}
```

Vsakemu dokumentu Cloudant je dodeljen unikatni identifikator (`_id`). Samodejno se doda tudi številka revizije (`_rev`), ki je formata *N-<zgoščena vrednost>*, kjer *N* predstavlja število, kolikokrat je bil dokument spremenjen. Kot kaže drugi primer, smo posodobili vrednost druge dejavnosti, ki jo opravlja Janez Novak. Poleg te vrednosti se je spremenila tudi vrednost `_rev`, kjer je *N* sedaj 2, namesto 1. Polja v dokumentih so lahko številke, nizi, logične vrednosti, objekti, polja, datumi, priponke (video, slika) in podobno. Pri številu elementov ali velikosti dokumenta ni nobenih omejitev.

Kot je značilno za večino nerelacijskih podatkovnih baz, tudi baza Cloudant nima sheme, ki bi določala strukturo vsebine dokumentov. Brez težav lahko dodajamo nove tipe dokumentov ali pa spreminjamo vsebino že obstoječih dokumentov, brez potrebne migracije podatkov kot pri SQL bazah. Do dokumentov lahko dostopamo preko HTTP z uporabo APIja za spletne storitve RESTful. Indeksi so zgrajeni s pomočjo funkcij MapReduce, ki omogočajo hitro poizvedovanje in analiziranje.

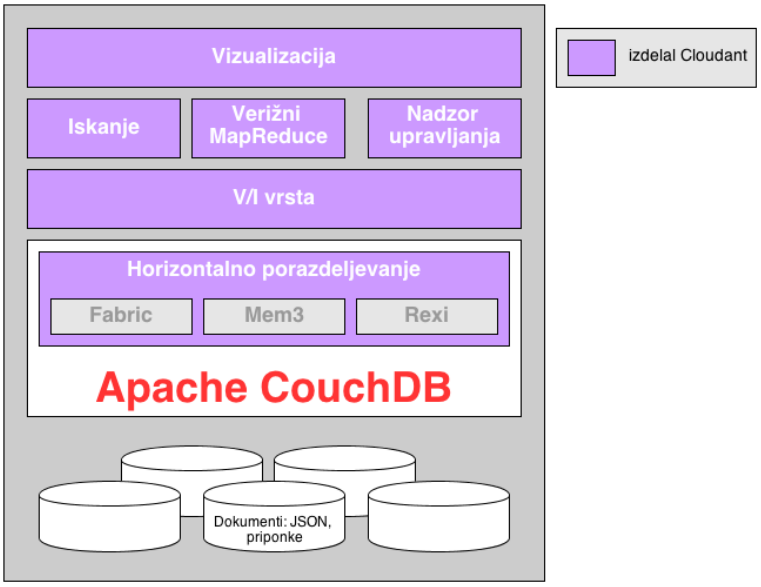
Značilnosti Ker baza CouchDB ni podpirala sposobnosti horizontalnega porazdeljevanja, ga je Cloudant implementiral sam v jeziku Erlang. Horizontalno porazdeljevanje se ukvarja z usmerjanjem in usklajevanjem interakcij podatkovne baze, usklajevanjem porazdeljenih poizvedb in podpira klic za oddaljeni postopek, ki omogoča boljše rezultate in odpornost na napake na oddaljenih vozliščih.

Za uspešno ravnanje z več kot milijardo interakcij na dan za več tisoč podatkovnih baz, Cloudant uporablja vhodno/izhodno vrsto (v nadaljevanju IOQ) za določanje prioritete zahtevam. IOQ analizira prioriteto vsake IO zahteve in jo servisira tako, da vsak najemnik dobi pravičen delež virov in ni nihče oškodovan. Tudi pri eno-najemništvu se uporablja IOQ, kjer lahko na nadzorni plošči sami določimo prioritete določenim interakcijam.

Vgrajeno imajo tudi iskanje, ki temelji na iskalni knjižnici Apache Lucene, ki podpira celovito tekstovno iskanje za zapletenejše poizvedbe.

Cloudant pri pisanju ali branju dokumentov uporablja konsistenten zgoščevalen algoritem za izračun dokumentove lokacije. Funkcije MapReduce in druge poizvedbe so izvedene tako, da se pošlje koda vsem vozliščem v gruči za vzporedno izvajanje, nato pa se rezultate vozlišč uredi z zlivanjem (ang. *merge sort*), da dobimo logičen pogled podatkovne baze [28].

Arhitektura Podatkovna plast Cloudant je sestavljena iz zbirke gruč podatkovnih baz, ki gostujejo in so porazdeljene čez več podatkovnih centrov po vsem svetu. Pri aktivaciji računa lahko uporabnik sam izbere fizično platformo (SoftLayer, AWS, Joyent, Microsoft Azure, Rackspace) in geograf-



Slika 3.2: Zgradba vozlišča Cloudant

sko območje (ZDA, Evropa, Azija), ki mu najbolj odgovarja, za čim boljšo kakovost storitve (npr. zniževanje omrežne latence).

Cloudant porazdeljuje podatkovne baze čez več gruč strežnikov in avtomatsko obravnava izravnavo obremenitve, upravljanje gruč, varnostno kopiranje in izvajanje porazdeljenih poizvedb. Privzeto so podatki shranjeni v več-najemniški (ang. *multi-tenant*) gruči. Več-najemništvo pomeni, da eno instanco podatkovne baze uporablja več t.i. najemnikov. Te podatkovne baze so ustrezno zavarovane in niso dostopne drugim najemnikom, razen če to eksplicitno dovoljujemo. Cloudant ponuja tudi zasebne, eno-najemniške gruče, ki prav tako porazdeljujejo in replicirajo podatke čez več strežnikov, vendar proti plačilu.

CAP teorem Cloudant zagotavlja razpoložljivost in particijsko toleranco. To zagotavljajo tako, da različna vozlišča v gruči vsebujejo kopije delov podatkov, ki se med seboj tudi kopirajo, zato lahko v primeru odpovedi enega vozlišča, druga vozlišča prevzamejo njegovo nalogo. S tem je zagotovljena za-

kasnjena konsistentnost, vendar ne absolutna konsistentnost, saj mora preteči nekaj časa, da so vsi podatki zopet ažurni. Med vozlišči se uporablja replikacijska shema gospodar-gospodar, ki omogoča, da lahko vsako vozlišče bere in piše dele podatkov.

3.2.2.2 MongoLab - MongoDB

Podatkovna baza MongoDB je ena izmed najbolj razširjenih nerelacijskih podatkovnih baz, zato jo kot storitev ponuja veliko podjetij. Največji ponudniki MongoDB kot storitev so MongoHQ, Mongo Machine in MongoLab, slednjega bomo v tem diplomskem delu opisali podrobnejše. Med danimi ponudniki ni večjih razlik. Kot že ime Mongo, ki izhaja iz angleške besede *humoungous* (ogromen), pove, gre za bazo, ki je sposobna upravljati z ogromno količino podatkov. Cilji so bili ustvariti podatkovno bazo, ki bo dobro skalabilna, imela dobre performančne zmogljivosti in bo preprosta za uporabo.

Podatkovni model MongoDB je prav tako dokumentno usmerjena podatkovna baza in je primerna za shranjevanje JSON dokumentov, ki so v binarnem formatu BSON. Tudi MongoDB ne zahteva v naprej definirane sheme. Podatkovna baza je razdeljena na več delov, ki se imenujejo zbirke (ang. *collections*). Zbirk ni potrebno posebej ustvariti, saj so ustvarjene samodejno ob prvem zapisu dokumenta v zbirko.

Funkcije MongoDB ima že sama po sebi zmožnost horizontalnega skaliranja čez več strežnikov, kar doseže z replikacijo, ki kopira podatke na drug strežnik, ali pa s samodejno porazdelitvijo (ang. *sharding*), ki razdeli zbirko na več manjših delov. Obe lastnosti nakazujeta na visoko razpoložljivost baze [13].

Vgrajeno ima tudi podporo za indeksiranje in tekstovno iskanje, za hitro procesiranje pa uporablja funkcije MapReduce.

MongoLab nudi neprestano spremljanje delovanja posameznega strežnika

z orodjem MongoDB Monitoring Service in varnostno kopiranje.

Za dostop do podatkovne baze lahko uporabimo standardne gonilnike MongoDB v poljubnem jeziku ali pa MongoLabov REST API.

Arhitektura MongoLab nudi več platform: Amazon, Google, Joyent, Rackspace in Widows Azure. Na voljo je več naročniških načrtov, kjer izbiramo med deljenim (ang. *shared*) in namenskim (ang. *dedicated*) gostovanjem. Pri deljenemu načrtu lahko izbiramo med:

- peskovnikom (ang. *sandbox*) - podatkovna baza na deljenem strežniku in na deljenem navideznem stroju,
- enim vozliščem - podatkovna baza na namenskem strežniku in na deljenem navideznem stroju,
- gručo vozlišč - podatkovna baza na namenskem strežniku na deljenem navideznem stroju. Zaradi replikacije nudi visoko razpoložljivost in samodejni preklop v primeru napake.

Pri namenskem načrtu pa imamo na voljo:

- namenske gruče - neomejeno število podatkovnih baz na namenskih strežnikih in na več namenskih navideznih strojih. Nudi visoko razpoložljivost in samodejni preklop v primeru napake ter namenske nastavitve za prioritete IOQ [29].

CAP teorem MongoDB zagotavlja particijsko toleranco, drugo lastnost pa lahko izbiramo. Med vozlišči se uporablja replikacijska shema gospodar-suženj, kjer je le eno vozlišče gospodar, ki lahko odobri pisanje podatkov. Druga vozlišča so sužnji, ki replicirajo podatke od gospodarja, da lahko opravljajo operacije branj. Zato pri zagotavljanju druge lastnosti ločimo dva scenarija:

- Nastavitve za pisanje (ang. *Write Concerns*) - določimo, kdaj je bilo pisanje uspešno (od ignoriranja napake do števila posameznih vozlišč,

ki so priznale pisanje). Torej bo pisanje neuspešno, če bo število potrebnih vozlišč manjše od trenutno dosegljivih vozlišč, s čimer škodujemo razpoložljivosti.

- Nastavitve za branje (ang. *Read Preferences*) - določimo, ali je branje možno le od gospodarja ali tudi od sužnjev. Če je branje možno tudi od sužnjev, s tem škodujemo konsistentnosti, saj če poteka branje samo pri gospodarju, ostala vozlišča še nimajo enakih podatkov [35].

3.2.3 Podatkovne baze s širokimi stolpci

Podatkovne baze s širokimi stolpci imajo zgradbo, ki je podobna relacijskim in ključ-vrednost podatkovnim bazam. Kot celota so shranjeni podatki enega stolpca in ne podatki ene vrstice, kot so shranjeni pri relacijskih podatkovnih bazah. Dodajanje novih stolpcev v bazo se naredi na osnovi vrstica za vrstico. Vsaka vrstica ima lahko poljubno veliko stolpcev, tudi nobenega. To omogoča, da ne povzročamo dodatnih stroškov, ki bi nastali, če bi morali dodati ničelne (ang. *null*) vrednosti. Za razliko od relacijskih, ki so usmerjene v vrstice, so stolpične podatkovne baze, kot že ime pove, usmerjene v stolpce. Vsako enoto lahko obravnamo kot par ključ-vrednost, enoto pa identificiramo s primarnim ključem. Posamezne enote shranjujemo v družine stolpcev, na primer v družino stolpcev `naslov`, bi shranili stolpce `ulica`, `poštna številka` in `kraj`. Družine stolpcev moramo običajno vnaprej opredeliti, medtem ko stolpcev ni potrebno. Shranjujemo lahko poljubne podatkovne tipe, dokler obstajajo kot polja bitov [12, 13].

Predstavniki podatkovnih baz so: Cassandra [36], HBase [37], Hypertable [38], Amazon SimpleDB [39], Google BigTable.
Storitve pa nudijo: Amazon SimpleDB [39], Instaclustr (Cassandra) [40], Google Cloud Storage (BigTable) [41]. Odločili smo se, da predstavimo še drugo Amazono storitev SimpleDB, ki je ena izmed prvih nerelacijskih podatkovnih baz kot storitev. Za razliko od storitve Instaclustr omogoča brezplačni paket. Google Cloud Storage pa je trenutno še v poznem beta razvoju.

3.2.3.1 Amazon SimpleDB - SimpleDB

Amazon poleg DynamoDB nudi tudi SimpleDB, ki je njegova predhodnica. Gre za podatkovno bazo, napisano v jeziku Erlang, ki sodi v družino stolpčnih baz. Je visoko razpoložljiva, prilagodljiva, vendar ima omejitve pri kapaciteti. Če naši podatki prekoračijo obseg omejitve velikosti domene (10 GB) ali omejitve pri poizvedovanju, lahko naredimo nove domene in tako razbremenimo prvotno domeno. Porazdeljevanje podatkov čez več domen pohitri naše operacije, saj različne domene uporabljajo različne vire, zato lahko nekatere operacije izvajajo paralelno in ne zaporedno. Vendar moramo porazdeljevanje podatkov opraviti sami. Omejitev obsega velikosti domene so odpravili pri Amazon DynamoDB, ki nima omejitev. Kljub ne popolnoma skalabilni podatkovni bazi, se dobro odreže pri manjšem obsegu, ki zahteva prilagodljivost pri poizvedovanju, kar doseže s samodejnim indeksiranjem atributov.

Podatkovni model Amazon SimpleDB ima hierarhičen podatkovni model, katerega korenski element je račun (ang. *account*). Vsak račun ima lahko eno do sto domen (privzeta vrednost, Amazon nam lahko priskrbi več domen), v katerih so organizirani podatki. Domene so sestavljene iz enot, enote pa iz parov atribut-vrednost. Po relacijskem modelu atributi zasedajo strukturo stolpca, vrednosti pa celico. En atribut ima lahko nič ali več vrednosti, indeksiranje pa je samodejno. Kot vidimo v Tabeli 3.1, lahko v eno domeno shranjujemo poljubne elemente, saj so atributi lahko brez vrednosti. Shranjujemo lahko nize znakov, ki so kodirani z UTF-8, katerih dolžina znakov ne presega 1024 bajtov [42].

ID	Kategorija	Ime	Barva	Velikost	Model
izd_01	Oblačila	Majica	Modra, Rdeča	S, M, L	
izd_02	Posoda	Ponev	Črna		M15
izd_03	Telefon	Galaxy	Črna		S4
izd_04	Avto	Audi			A1

Tabela 3.1: Primer domene.

Funkcije Amazon SimpleDB ponuja enostaven API, ki sledi REST in SOAP smernicam ter je na voljo kot spletna storitev, ki je narejena za enostavno interakcijo s podatki. Je visoko razpoložljiva, saj samodejno porazdeli kopije podatkov na različne geografske lokacije, ki jih določimo sami. Izberemo lahko takšno lokacijo, ki bo blizu naših uporabnikov, kar bo zmanjšalo zakasnitev.

Ker nima vnaprej predpisane sheme, lahko enostavno dodajamo nove atribute, ko je to potrebno.

Podobno kot pri DynamoDB lahko izbiramo kakšno vrsto konsistentnosti branja želimo [11].

Arhitektura Amazonova infrastruktura sestoji iz več deset tisoč strežnikov, ki se nahajajo v številnih podatkovnih centrih po svetu.

CAP teorem Amazon SimpleDB zagotavlja visoko razpoložljivost in particijsko toleranco. Ob vsakem shranjevanju podatkov se ustvarijo replike v različnih podatkovnih centrih izbrane regije, kar zagotavlja visoko razpoložljivost in particijsko toleranco v primeru odpovedi sistema. S tem so se odpovedali konsistentnosti in zagotavljajo le zakasnjeno konsistentnost.

3.2.4 Graf podatkovne baze

Graf podatkovne baze shranjujejo podatke, ki so lahko predstavljeni kot grafi. Poznamo več vrst grafov: neoznačeni grafi, neusmerjeni grafi, hipergrafi in podobno. Kot graf se lahko razlaga tudi podatkovni model RDF (ang. *Resource Description Framework*). RDF je v splošnem metoda za izražanje znanja in je temelj semantičnega spleta, ki je nastal z namenom predstavitve porazdeljenih in strukturiranih podatkov v obliki, primerni za interpretacijo računalnikov [43]. Podatkovni model RDF ima obliko trojčka (ang. *triple*), ki je sestavljen iz osebka, predikata in predmeta (ang. *subject, predicate, object*), podobno kot je sestavljen stavek. Stavek "Miza je modra" lahko razdelimo na osebek : miza, predikat : je, objekt : modra [44]. Zaradi nji-

hove zgradbe, kjer si predikat lahko predstavljamo kot povezavo med dvema objektoma, se graf podatkovne baze uporabljajo pri socialnih omrežjih. Vrsto graf podatkovne baze, ki lahko shranjuje trojčke, imenujemo triplestore. Najbolj razširjen poizvedovalni jezik, s katerim lahko poizvedujemo po triplestore podatkovnih bazah, se imenuje SPARQL.

Predstavniki graf in triplestore podatkovnih baz: Neo4j [45], HyperGraphDB [46], InfiniteGraph [47], Apache Jena [48], SparkleDB [49] in Dydra [50]. Slednja podatkovna baza je na voljo kot storitev in jo bomo podrobneje obravnavali v nadaljevanju.

3.2.4.1 Dydra

Dydra nudi triplestore graf podatkovno bazo kot storitev. Trenutno je njihova storitev še v zaprti beta različici, zato so informacije in dokumentacija o storitvi zelo omejene. Performančno se najbolje izkaže na primerih uporabe, ki temeljijo na grafih, na primer socialna omrežja. Podobno kot ostale graf podatkovne baze, tudi Dydra uporablja poizvedovalni jezik SPARQL. Temelji na SPARQL mehanizmu za poizvedovanje (ang. *query engine*), ki se imenuje SPOCQ, ki je napisan v programskem jeziku Lisp. RDF shramba je napisana v programskem jeziku C s knjižnico Raptor, ki serializira trojčke.

3.2.4.2 Podatkovni model

Dydra za shranjevanje podatkov uporablja prej omenjeni podatkovni model RDF. Podatkovni model RDF je izjemno prilagodljiv s številnimi serializacijami. Serializacija je proces pretvarjanja stanja objekta v format, ki je primeren za shranjevanje (npr. v datoteko). Dydra trenutno podpira naslednje datotečne formate: Turtle, TriG, N-Triples, N-Quads, RDF/JSON, RDF/XML, RDFa. Podrobneje bomo predstavili format RDF/XML, ki omogoča predstavitev RDF trojčkov v obliki XML in je podprt s strani večine aplikacij na semantičnemu spletu. V korenski znački `rdf:RDF` je več opisnih značk `rdf:Description`. Z attributom `rdf:about` določimo osebek vsem izrazom v opisanem elementu. Predmet pa je predstavljen z značko `rdf:resource`

ali pa kot literal znotraj oznak predikata [51].

Primer RDF/XML trojčka:

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     <rdf:Description rdf:about="osebek">
          <predikat rdf:resource="predmet" />
          <predikat>literal</predikat>
     </rdf:Description>
</rdf:RDF>
```

Podatke lahko vnašamo v repozitorij tudi preko spletne strani, kjer imamo na voljo dve možnosti: naložimo lokalno datoteko ali pa podamo URL, kjer se nahajajo podatki.

Funkcije Dydra je visoko skalabilna podatkovna baza. Ker je storitev še v beta različici, se moramo dogovoriti v okvirnem številu trojčkov, ki jih bomo shranjevali, vendar načeloma ni omejitev. Dydra samodejno skrbi za visoko zmogljivost, verzioniranje podatkov in obnovo podatkov v primeru nesreče. Uporablja odprte standarde, priporočene s strani W3C, zato je izjemno prilagodljiva. Trenutno lahko navaden uporabnik izvaja 5 poizvedb na sekundo in shranjuje 1 milijion trojčkov, vendar se lahko dogovorimo za večje limite.

Arhitektura Dydra je več-najemniška podatkovna baza. Trojčki so samodejno razdeljene po več vozliščih v gručah.

CAP teorem Ni podatkov.

Poglavje 4

Primerjava ponudnikov

4.1 Kriteriji za ocenjevanje

Na voljo imamo kar nekaj ponudnikov podatkovnih baz kot storitev, zato je težko izbrati pravega. Ponujajo različne rešitve, ki so zelo odvisne od podatkovne baze, na kateri temeljijo. Za ocenjevanje najprimernejše podatkove baze kot storitve je na voljo veliko različnih pristopov. V tem diplomske delu smo za kriterije izbrali funkcionalnosti, ki jih ponujajo, ter ceno, podporo uporabnikom in načine dostopa. Velja pa omeniti tudi kategorijo podatkovne baze, ki je načeloma glavni dejavnik izbire podatkovne baze, saj se podatkovne baze različno izkažejo pri reševanju določenih problemov.

4.1.1 Kategorija podatkovne baze

Opis kategorij podatkovne baze smo obravnavali že v drugem poglavju. V tem diplomskem delu smo se osredotočili na štiri kategorije, in sicer ključ-vrednost, dokumentno usmerjene, podatkovne baze s širokimi stolpci in graf podatkovne baze. Izbira kategorije podatkovne baze je zelo pomemben dejavnik, saj se nekatere v določenih problemih obneseje bolje kot druge in obratno. V nadaljevanju bomo navedli primere uporabe za določene kategorije podatkovnih baz.

Ključ-vrednost podatkovna baza Kot smo že omenili, je zgradba te podatkovne baze zelo enostavna. Enostavnost pa postane težava, kadar želimo izvesti zapletenejšo obdelavo podatkov, saj nimajo dobre podpore za indeksiranje. Dobro rešujejo probleme, kjer se poleg ključa uporablja le ena vrednost ali pa seznam vrednosti, in kjer podatki med seboj niso povezani. Seznami vrednosti so lahko na primer najbolj prodajani izdelki, kategorije izdelkov ali vsebina nakupovalnega vozička. Ne obnesejo pa se dobro, če moramo poiskati ključe glede na vrednost atributov. Amazon DynamoDB je zadnjo slabost rešil z uporabo dodatne storitve Amazon Cloud Search. Primeri uporabe:

- Shranjevanje sej uporabnikov, shranjevanje podatkov o nastavitvah uporabnikov
- Nakupovalni voziček
- Enostavna analitika - Electronic Arts [52], mednarodno podjetje za razvoj videoiger uporablja Amazonovo storitev DynamoDB za prikazovanje statistike v realnem času. IMDb (Internet Movie Database) [53] prav tako uporablja DynamoDB za ocenjevalne lestvice filmov.
- Shranjevanje metapodatkov - SmugMug [54], spletna stran za gostovanje slik uporablja DynamoDB za shranjevanju metapodatkov o slikah in videih za milijone uporabnikov. Mnoga podjetja shranjujejo tudi metapodatke o internetnih publikacijah.

Dokumentno usmerjene podatkovne baze Dokumentno usmerjene podatkovne baze so zaradi svoje zgradbe dobre za shranjevanje podatkov, katerih oblike ne poznamo v naprej. Uporabljamo jih v primerih, ko želimo izvajati poizvedbe nad več atributi. Ponudnika, ki ponujata dokumentno usmerjeno podatkovno bazo se razlikujeta v tem, da Cloudant (CouchDB) shranjuje dokumente JSON, MongoLab (MongoDB) pa dokumente BSON. BSON (ang. Binary JSON) je razširjena različica JSON-a, ki lahko poleg enakih tipov kot JSON, z izjemo univerzalnega tipa števila, shranjuje tudi tipa datum in bitna tabela. Glavna prednost formata BSON pred JSON je njegov

izkoristek, tako prostorski kot računski, zaradi binarnega formata. Vendar so te razlike med oblikama dokumentov tako majhne, da niso ključnega pomena pri izbiri med tema podatkovnima bazama. Primeri uporabe:

- Blogerske platforme za shranjevanje uporabniških komentarjev, registracij, uporabniških profilov - Overdriver [55], igričarska platforma v oblaku, uporablja MongoLab za socialne funkcionalnosti spletne skupnosti razvijalcev iger.
- Analitika v realnem času - Podjetje Meteor Solutions [56], ki ugotavlja ciljno publiko za podjetja, uporablja ponudnika Cloudant za statistično analizo v realnem času.
- Seznam opravil - Orchestra [57] uporablja Cloudant za razvoj iOS aplikacije za seznam opravil.
- E-nakupovanje, beleženje dogodkov, koledar

Baze s širokimi stolpci Podatkovne baze s širokimi stolpci se dobro obnesejo v primerih pol strukturiranih podatkov, ko želimo entitete pogrupirati po skupnih lastnostih, čeprav se njihove karakteristike razlikujejo. Za primer lahko vzamemo izdelek z lastnostjo Velikost. Nekateri izdelki imajo lahko velikost podano s širino, višino in globino, drugi s premerom, tretji samo z dolžino. Ne obnesejo pa se dobro, če ne vemo točno kako bomo uporabljali podatke.

Primeri uporabe za podatkovne baze s širokimi stolpci so podobni tistim pri dokumentno usmerjenih podatkovnih bazah.

Graf podatkovne baze Graf podatkovne baze se najbolje izkažejo za shranjevanje močno povezanih podatkov.

Primeri uporabe:

- Socialna omrežja
- Povezave med filmi

- Lokacijske storitve
- Priporočjivost

4.1.2 Sekundarno indeksiranje

Indeks je podatkovna struktura, ki omogoča učinkovitejše poizvedovanje. Ker primarno indeksiranje ne zadostuje vsem našim potrebam poizvedovanja, so mnoge podatkovne baze vpeljale sekundarno indeksiranje. Sekundarno indeksiranje omogoča indeksiranje po atributu, ki ni primarni ključ in omogoča iskanje podatkov, sortiranje in agregacije. Namesto sekundarnega indeksiranja lahko uporabimo algoritm MapReduce, ki je sestavljen iz dveh funkcij, in sicer `map`, ki obdela dokument in vrne vmesne pare tipa ključ/vrednost, ter `reduce`, ki združi pridobljene pare tipa ključ/vrednost v eno samo vrednost za pripadajoč ključ [58].

Amazon DynamoDB Amazon DynamoDB podpira možnost uporabe lokalnih sekundarnih indeksov. To so podatkovne strukture, ki imajo še vedno enak zgoščevalni ključ, vendar drug razponski ključ. Definiramo lahko le 5 takih indeksov na tabelo. Kreirani so lahko le ob kreiranju tabele, kasneje jih ne moremo spremeniti ali izbrisati. Sekundarni indeks poleg zgoščevalnega in razponskega ključa vsebuje še kopije drugih atributov, ki jih določimo sami. Tako hitro dostopamo do teh podatkov. Lahko izvajamo tudi poizvedbe, ki vračajo atribute, ki so skopirani v ključu, vendar bodo te počasnejše in bodo porabile več enot pretoka. Podatki so organizirani po indeksovem razponskem ključu. Lokalni indeksi so samodejno posodobljeni ob spremembi tabele. Shranimo lahko do 10GB podatkov na eno vrednost zgoščenega ključa.

Cloudant Cloudant ustvari primarni indeks samodejno, zapiše se kot element `_all_docs`, ki vrne id, ključ in vrednost za vsak dokument v podatkovni bazi. Ker tako generičen primarni indeks ni dovolj, imamo na voljo API možnosti za oblikovanje tega ključa. Podpira tudi sekundarne indekse, ki jih definiramo z uporabo MapReduce. Sekundarni indeksi oz. pogledi,

kot se imenujejo pri CouchDB, so definirani s funkcijo `map`, ki izbere ustrezne podatke iz dokumentov in opcijsko funkcijo `reduce`, ki agregira izbrane podatke. Ima tudi možnost verižne uporabe funkcije MapReduce, ki dovoljuje nadaljnjo uporabo funkcije MapReduce nad rezultatom, ki ga vrne ta funkcija. Indeksi uporabljajo B^+ drevesno strukturo.

MongoLab Vsi indeksi pri MongoDB so sekundarni indeksi. MongoDB podpira indekse na vseh elementih ali ugnezdenih elementih v dokumentu. Ustvarimo lahko tudi spojne (ang. *compound*) indekse, kjer en indeks hrani reference do več elementov. Če indeksiramo element, ki vsebuje polje, potem se ustvari večnivojski (ang. *multikey*) indeks, ki vsebuje vsak posamezen element polja. Uporablja B- drevesno podatkovno strukturo. Vsaka poizvedba uporabi le en indeks, ki se izbere samodejno na podlagi najboljše učinkovitosti.

Amazon SimpleDB Ne podpira uporabe sekundarnih indeksov.

Dydra Dydra ne uporablja sekundarnih indeksov.

Sklep Najslabše indeksiranje omogoča Amazon SimpleDB, saj nima podpore za sekundarne indekse. Tudi druga Amazonova storitev, DynamoDB se obnese slabše kot ponudnika dokumentno usmerjenih baz, saj je število sekundarnih indeksov omejeno. Cloudant in MongoLab oba ponujata naprednejše indeksiranje, s katerima zadostita zapletenejšim poizvedbam. Kot omenjeno, moramo za sekundarno indeksiranje pri Cloudantu pisati funkcije MapReduce, pri MongoLab pa uporabljamo ukaz `ensureIndex`, ki generira različne indekse.

4.1.3 Geoprostorsko indeksiranje in poizvedovanje

V zadnjem času so v porastu aplikacije, ki uporabljajo geoprostorske podatke, kot so geografska širina in geografska dolžina ter geoprostorske podatkovne

tipe kot so točke, premice, mnogokotniki in elipse. Tipična poizvedba aplikacije, ki uporablja geoprostorske podatke, je na primer: "Najdi vse restavracije v radiju petih kilometrov od moje trenutne lokacije".

Amazon DynamoDB V času pisanja diplomskega dela Amazon DynamoDB še ni podpiral geoprostorskega indeksiranja, vendar je Amazon napovedal, da bo podporna knjižnica kmalu razvita.

Cloudant Cloudant ima podporo za dvo in tri dimenzionalne geoprostorske indekse. Podpira shranjevanje vseh referenčnih koordinatnih sistemov in izvaja poizvedbe bližine, presečišč, prostorskega obsega (škatla, mnogokotnik, radij, elipsa) in časovne analize. Za ustvarjanje geoprostorskih indeksov se uporabljajo napredna R*- drevesa.

MongoLab MongoDB ima podporo za dvo dimenzionalne geoprostorske indekse in podatkovne tipe točka, premica in mnogokotnik. Uporablja B-drevesne podatkovne strukture za vse indekse.

Amazon SimpleDB Amazon SimpleDB nima vgrajene podpore za geoprostorsko indeksiranje in poizvedovanje.

Dydra Trenutno Dydra še nima vgrajene podpore za geoprostrosko indeksiranje in poizvedovanje, vendar je le-ta v razvoju.

Sklep Če potrebujemo rešitev, ki bo podpirala geoprostorske podatke, lahko izločimo obe Amazonovi storitvi in Dydro. Cloudant omogoča boljšo podporo, saj v primerjavi z MongoLabom podpira tudi tri dimenzionalne indekse. Prav tako so R*- drevesa bolj učinkovita kot B-drevesa.

4.1.4 Celovito tekstovno iskanje

Celovito tekstovno iskanje je tehnika, ki omogoča iskanje nizov v zbirkah in dokumentih.

Amazon DynamoDB Amazon DynamoDB sam po sebi nima vgrajene podpore tekstovnemu iskanju. Za to lahko uporabimo dodatno Amazono storitev Cloud Search, vendar ni vključena v samo storitev AmazonDynamoDB.

Cloudant Cloudant ima vgrajeno odprtokodno knjižnico Apache Lucene, napisano v Javi, funkcionalnost pa so poimenovali Lucene Search 2.0. Za iskalna gesla sprejema vrednosti naslednjih tipov: boolean, nizi, števila (64-bitna števila s plavajočo vejico). Ne podpira pa iskanja objektov, polj, nedifiniranih in ničnih vrednosti. Iskanje je zelo podobno kot pisanje funkcij MapReduce, le da zamenjamo ključno funkcijo, ki je namesto `emit()` sedaj `index()`.

MongoLab Podpora celovitemu tekstovnem iskanju je bila dodana v zadnji verziji MongoDB, to je verzija 2.4. Dodali so nov indeks z imenom `text` in istoimenski ukaz. Med izvrševanjem ukaza `text` in kreiranjem indeksa, tokenizira in obreže iskalno geslo do korena (ang. *stemming*). Iskanje poteka tako, da se določajo točke vsakemu dokumentu, ki vsebuje iskalno geslo v indeksiranih poljih. Te točke določajo ustreznost dokumenta glede na iskalno geslo - večje kot je število točk, bolj ustrezen je rezultat.

Amazon SimpleDB SimpleDB nima vgrajene podpore tekstovnemu iskanju.

Dydra Trenutno še ne podpira celovitega tekstovnega iskanja.

Sklepne ugotovitve Zopet se Cloudant in MongoLab izkažeta kot najboljša ponudnika.

4.1.5 Cena

Ponudniki se poslužujejo različnih načinov kako zaračunavati storitve. Prav cena storitve je po navadi eden izmed glavnih kriterijev pri ocenjevanju sto-

ritve.

Amazon DynamoDB Amazon DynamoDB svoje storitve zaračunava glede na več dejavnikov, vsem strankam pa nudi enake storitve, ki zajemajo sporazum o ravni storitve (ang. *Service-Level Agreement).*

Kapaciteta pretoka Število operacij pisanj in branj imenujemo tudi kapaciteta pretoka. Cene kapacitet pretoka se razlikujejo od območja, v katerem jih zakupimo. V povprečju znaša cena 0.0075$ na uro za vsakih 10 enot pisanja (dovolj kapacitet do 36 000 pisanj na uro) in 0.0075$ na uro za vsakih 50 enot branja (dovolj za 180 000 konsistentnih branj ali dvakrat toliko zakasnjeno konsistentnih branj). Ena pisalna enota nam omogoča eno pisanje elementa do velikosti 1KB. Če prekoračimo velikost 1KB, bomo za vsak nadaljni KB porabili eno pisalno enoto, zaokroženo na najbližje celo število. Torej, če želimo zapisati 10 elementov na sekundo, velikosti 2.4KB, bomo porabili $10\times2.4=24$ pisalnih enot. Ena bralna enota nam omogoča eno konsistentno branje ali dve zakasnjeno konsistentni branji izdelkov do velikosti 4KB. Če prekoračimo 4KB, bomo za vsake nadaljne 4KB porabili eno bralno enoto, zaokroženo na najbližje število. Torej, če želimo brati 10 elementov na sekundo, velikosti 2.4KB, bomo porabili $10\times(\frac{2.4}{4})=6$ bralnih enot.

Shramba podatkov in indeksov Velikost podatkov, ki jih zapišemo v bazo ni enaka velikosti baze, saj se ustvarjajo še indeksi, ki zasedajo dodaten prostor. Amazon zaračunava storitev tudi glede na porabljen prostor, kar v povprečju znaša 0.2885$ na GB podatkov.

Prenos podatkov Amazon zaračunava prenos podatkov iz Amazon DynamoDB. Če se podatki prenašajo med storitvami Amazon v enaki regiji, ni dodatnih stroškov.

Zakupljene količine Amazon ponuja tudi nakup vnaprej zakupljenih količin za eno ali tri leta. Vnaprej plačamo določeno ceno za število enot branja

Prenos (OUT) / mesec	Cena	Prenos (OUT)	Cena
Prvih 1 GB	0.000$	Naslednjih 350 TB	0.050$
Do 10 TB	0.120$	Naslednjih 524 TB	Po dogovoru
Naslednjih 40 TB	0.090$	Naslednjih 4 PB	Po dogovoru
Naslednjih 100 TB	0.070$	Več kot 5 PB	Po dogovoru

Tabela 4.1: Cene zunanjega prenosa podatkov za regijo US East (N. Virginia) na mesec.

in pisanja ter potem mesečno plačujemo porabo, vendar po znižani ceni. Tako lahko zakupimo 5000 enot pisanj, ki jih vnaprej plačamo v povprečju 11250$, ter nato plačujemo v povprečju 0.74$ na uro. Na ta način lahko prihranimo okoli 48% več kot, če bi plačevali kolikor dejansko porabimo. Podobne rezultate dobimo pri enotah branja.

Brezplačen paket Amazon DynamoDB vsak mesec nudi brezplačen paket, ki vsebuje 100 MB shrambe in kapaciteto pretoka 5 pisanj na sekundo in 10 branj na sekundo. S tem lahko opravimo 432 000 pisanj in 864 000 branj na dan brezplačno.

4.1.5.1 Cloudant

Kot že omenjeno, Cloudant ponuja dve možnosti shranjevanja podatkov po gručah, in sicer več-najemništvo in namensko shranjevanje.

Več-najemništvo Ponuja neomejeno število podatkovnih baz, zahtev HTTP na sekundo in shranjevanje. Na voljo nam je nadzorna plošča, v kateri lahko brskamo in spreminjamo podatkovne baze preko spleta.

Shranjevanje podatkov	1.00$ / GB / mesec
Težje zahteve HTTP (PUT, POST, DELETE)	0.015$ na 100 zahtev
Lažje zahteve HTTP (GET, HEAD)	0.015$ na 500 zahtev

Tabela 4.2: Postavke zaračunavanj pri več-najemništvu.

Kot vidimo v zgornji tabeli, Cloudant zaračunava le toliko, kolikor porabimo, ne vsebuje nobenega mesečnega paketa. Loči med zahtevnostjo zahtev HTTP, ki jih lahko poenostavljeno razdelimo na branje, kot lažje operacije, in pisanje, kot zahtevnejše operacije. Če je naša skupna mesečna poraba manjša kot 5.00$, nam Cloudant uporabljenih storitev ne bo zaračunal.

Primer brezplačne uporabe storitve Vzemimo za primer, da naša aplikacija opravi dvakrat več branj kot pisanj. Za manj kot 5.00$ lahko tako shranimo do 500 MB podatkov in naredimo približno 22000 pisanj in 44000 branj na mesec.

**Primer uporabe za plačilo 15$** Uporabimo lahko njihove storitve v vrednosti 20$, saj je prvih 5$ uporabe brezplačnih. Tako lahko ob enaki predpostavki kot v prejšnem primeru, shranimo do 1.984GB podatkov in naredimo približno 87000 pisanj in 174000 branj na mesec

Namensko shranjevanje Poleg storitev, ki jih nudi več-najemništvo zajema tudi sporazum o ravni storitve, ki zagotavlja določeno kakovost storitve glede števila sočasnih uporabnikov, indeksiranja in zahtev HTTP. Na voljo nam je specifično orodje na nadzorni plošči, ki določa prioritete IOQ. V primeru težav imamo na voljo prednostno podporo, kar bomo obravnavali v nadaljevanju.

Cene namenske gruče se gibljelo od okoli 2700$ na mesec za gruče treh vozlišč, pa vse do 5500$ na mesec za najbolj robustne namenske storitve.

4.1.5.2 MongoLab

MongoLab ponuja več platform, na katerih lahko gostujemo in katerih izbira vpliva na ceno storitve. Nudijo nam gostovanje na platformah Amazon, Google, Joyent, Rackspace in Windows Azure. Storitve zaračunavajo na podlagi naročitvenih načrtov, ki se delijo na deljeno in namensko gostovanje.

Deljeni načrti Deljeni načrti nudijo nadzorno ploščo, REST API in v obeh plačniških naročninah tudi MongoDB sistem za nadzorovanje (ang. *MongoDB Monitoring System*) ter dnevnik beleženja dostopov v realnem času in v preteklosti.

Pri deljenem načrtu imamo možnost treh naročnin, katerih glavna razlika je kam se shranjujejo podatki, kot smo že omenili v poglavju 3.4.2. Mesečne naročnine so vnaprej zakupljene količine podatkovnih shramb. Če prekoračimo osnovno vključeno količino shrambe, lahko do neke kapacitete dokupujemo shrambo. Torej plačujemo le količino shranjenih podatkov in ne števila operacij, ki jih izvajamo.

	Peskovnik	Eno-vozlišče	Gruča (2 vozlišči)
Vključena shramba	0.484 GB	1.984 GB	1.984 GB
Mesečna naročnina	Brezplačno	15.00$	89.00$
Maks. shramba	0.484 GB	8 GB (5.00$/GB)	8GB (5.00$/GB)
Maks. naročnina	Brezplačno	45.00$	149.00$

Tabela 4.3: Cenik pri deljenem načrtu.

Namenski načrt Poleg storitev, ki jih dobimo pri deljenih naročninah, namenski načrt nudi neomejeno shrambo, število podatkovnih baz ter zagotovljen RAM predpomnilnik. V vsak načrt je vključena ena varnostna kopija na dan. Lahko imamo tudi večkratno varnostno kopiranje, vendar proti plačilu. V primeru težav nam je na voljo neprestana podpora preko elektronske pošte za nujne primere.

Podobno kot pri deljenih načrtih, pri namenskih zakupimo mesečno shrambo. Na voljo imamo več paketov, od Mini pa vse do paketa 4X-Large, kjer se sami dogovorimo o potrebnih kapacitetah. Vsi paketi zagotavljajo določeno mero RAM-a, ki bo vedno na voljo. Nekateri paketi vključujejo tudi možnost uporabe Amazonovih diskov SSD. Znotraj vsakega paketa je na voljo tudi razširjen paket, ki ob isti količina RAM-a omogoča večjo kapaciteto shranjevanja.

	Mini	X-Small	Small	Medium	Large	X-Large
RAM	1.7 GB	3.7 GB	7.5 GB	15 GB	34 GB	68 GB
Paket 1	200$	400$	850$	1600$	2800$	4800$
Shramba	40 GB	60 GB	120 GB	200 GB	300 GB	400 GB
SSD	Ne	Ne	Da(450$)	Da(450$)	Da(750$)	Da(750$)
Paket 2	260$	500$	1050$	1900$	3200$	5300$
Shramba	60 GB	150 GB	400 GB	600 GB	800 GB	1 TB
SSD	Ne	Ne	Da(550$)	Da(550$)	Da(850$)	Da(850$)

Tabela 4.4: Cenik pri namenskem načrtu pri gostovanju Amazon.

4.1.5.3 Amazon SimpleDB

Amazon SimpleDB storitev zaračunava glede na podobne dejavnike kot Amazon DynamoDB, le da namesto kapacitete pretoka računa uro, glede na izkoriščenost stroja.

Izkoriščenost stroja Amazon SimpleDB računa izkoriščenost stroja glede na dane zahteve. Vsaka posamezna zahteva porabi različno časa, odvisno od tipa zahteve, števila atributov in dolžine atributov. Ena računska ura pomeni, da bomo eno uro uporabljali en procesor 2007 1.7 Ghz Xeon. Cena ena računske ure v povprečju znaša 0.156$ in je, podobno kot ostale cene, odvisna od območja. Koliko časa je porabil procesor lahko vidimo v vsakem odgovoru, v parametru `BoxUsage`. Kreiranje in brisanje domene na primer porabi statične 0.0055590278 ure procesorskega časa, medtem ko za dodajanje in brisanje atributov velja formula:

$$0.0000219907 + 0.0000000002N^3 \tag{4.1}$$

, kjer je N število število atributov. Kot je razvidno iz formule, čas narašča kubično z vsakim dodanim atributom. Po mnenju nekaterih, formula ni odraz dejanskega časa računanja. Zaradi formule se nekatere zahteve izkažejo za veliko cenejše, če na primer zahtevo, ki zapiše 53 atributov razdelimo v dve ali več zahtev [59].

Shramba podatkov in indeksov Amazon zaračunava storitev tudi glede na porabljen prostor, ki v povprečju znaša 0.2788$ na GB podatkov. Porabljen prostor lahko izračunamo kot velikost naših podatkov, ki ji prištejemo 45 KB za vsak element, ime atributa in par atribut-vrednost.

Prenos podatkov Prenos podatkov je obračunan enako kot pri storitvi Amazon DynamoDB.

Brezplačen paket Podobno kot Amazon DynamoDB, tudi SimpleDB ponuja brezplačen paket. Paket vsebuje 25 računskih ur na mesec in 1 GB shrambe podatkov.

4.1.5.4 Dydra

Trenutno je storitev Dydra še brezplačna, vendar bo v prihodnosti plačljiva. Podatki o ceni storitve še niso na voljo.

4.1.5.5 Sklep

Najugodnejša mesečna brezplačna paketa nudi Amazon s storitvama Amazon DynamoDB in SimpleDB. Ker SimpleDB računa ceno glede na izkoriščenost stroja, ki je odvisna od vsake posamezne zahteve, lahko cena zelo varira, še posebej, če uporabimo zahteve za dodajanje veliko atributov. Zato bi med tema storitvama za ugodnejšo izbrali Amazon DynamoDB. Cloudant se izkaže za zelo dragega ponudnika, saj lahko v primerjavi z Amazon DynamoDB izvedemo bore malo operacij branja in pisanja. Če primerjamo prvi plačljivi paket MongoLaba, ki stane 15$ in primer porabe 15$ pri Cloudantu, vidimo, da se MongoLab paket cenovno obrestuje le, če naredimo več kot 87000 pisanj in 174000 branj. Ker MongoLab svoje storitve zaračunava le glede na količino shrambe podatkov, se izkaže za pametnejšo izbiro kot Cloudant, kadar shranjujemo malo podatkov in opravljamo veliko operacij. V obratnem primeru se bolj obrestuje izbira Cloudanta kot našega ponudnika podatkovne baze.

4.1.6 Podpora uporabnikom in skupnosti

Zelo pomemben kriterij pri storitvah, ki jih nudijo nove tehnologije, je vzdrževanje dobre dokumentacije ter učinkovita in hitra podpora uporabnikom. Ker gre za relativno nove tehnologije, je pomembno tudi, da so na voljo kratki vodiči, ki usmerjajo uporabnika pri uporabljanju storitve. Pri mnogih ponudnikih so se uporabniki združili v spletne skupnosti, ki odgovarjajo na zastavljena vprašanja na spletnih straneh, namenjenih reševanju programerskih problemov.

Amazon DynamoDB Amazon na spleti strani `http://aws.amazon.com/documentation/dynamodb/` nudi obsežno dokumentacijo, ki se deli na vodič za razvijalce in reference API. V vodiču za razvijalce je podrobno opisan proces, kako začeti z uporabo njihove storitve, kateremu so priložene tudi zaslonske maske. Na naslovu `http://aws.amazon.com/dynamodb/faqs/` najdemo tudi pogosto zastavljena vprašanja uporabnikov, ki so združena v smiselne področne sklope. Skupnost lahko postavlja vprašanja na namenskem spletnem forumu, kjer Amazon objavlja tudi pomembne novice. Ker Amazon nudi mnogo storitev, naslavlja skupnost kot celoto, ne le uporabnike Amazon DynamoDB. Tako lahko na kanalu Youtube `AWS Cloud` najdemo veliko videovsebin, povezanih z Amazonovimi storitvami v oblaku. Vprašanja lahko uporabniki postavljajo tudi na Twitterju na `@awscloud` in na Facebooku na `https://www.facebook.com/amazonwebservices`. Uporabniki se lahko včlanijo tudi v uporabniške skupine, kjer lahko komunicirajo z drugimi razvijalci. Uporabniške skupine so dostopne na naslovu `https://aws.amazon.com/usergroups/`. Nekaj vprašanj in odgovorov uporabnikov pa se nahaja na spletni strani `http://stackoverflow.com`. Za svoje uporabnike vsakoletno organizirajo tudi konferenco "AWS re:Invent". Amazon nudi več paketov podpore uporabnikom. Osnovni paket, ki je brezplačen, nudi podporo le glede težav z uporabniškim računom in plačevanjem, ne omogoča pa tehnične podpore. Na voljo so trije plačjivi paketi, ki so prikazani v tabeli 4.5.

	Developer	Business	Enterprise
Dostop do podpore	Email v delovnem času	Telefon, klepet, delitev zaslona (24/7)	Business in Tehnični upravljalec računa
Št. oseb, ki lahko kontaktirajo podporo	1	5	Neomejeno
Odvizni čas	<12 ur	<1 ura	<15 minut
Diagnostična orodja pri uporabniku	Da	Da	Da
Upravljanje poslovnih pregledov	/	/	Da
Cena	49$ na mesec	Odvisno od mesečne porabe	Odvisno od mesečne porabe

Tabela 4.5: Amazonovi paketi podpore uporabnikom.

Kot je razvidno iz tabele 4.5, Amazon v plačljivih paketih nudi napredno tehnično podporo, vendar proti plačilu, ki lahko v Business in Enterprise modelu naraste do več deset tisoč ameriških dolarjev.

Cloudant Cloudant ima za razvijalce na voljo dokaj obsežen vodič, ki vodi uporabnika čez praktične primere uporabe njihove storitve. Na naslovu `https://docs.cloudant.com/` je na voljo obsežna dokumentacija, ki se deli na dva dela, in sicer na reference API in na razlago nekaterih konceptov. Skupnost lahko vprašanja postavlja na spletni strani `http://stackoverflow.com`, kjer jih označi z značko `#cloudant`. Na označena vprašanja potlej odgovarjajo Cloudantovi strokovnjaki, pa tudi ostali člani skupnosti. Če želimo čim hitrejši odgovor, se lahko prijavimo na IRC kanal `#cloudant` na `http://webchat.freenode.net/`, kjer prav tako odgovarjajo strokovnjaki in

člani skupnosti. Na voljo so tudi na Twitterju na `@cloudant`, kjer redno odgovarjajo na zastavljena vprašanja in na elektronski pošti, kjer odgovorijo v roku enega delovnega dneva. Če uporabnik vzamem namenski načrt ali pa, če stroški presegajo 500$ na mesec, mu je na voljo prednostna podpora, ki zagotavlja email odgovor v štirih urah. Prav tako so mu za vsa vprašanja na voljo preko telefona. Svojim uporabnikom nudijo tudi nekaj ur brezplačnih konzultacij, kasnejše konzultacije pa se plačuje. Ker je projekt odprtokoden, so na voljo tudi na `https://github.com/cloudant`. Okvirno enkrat na mesec organizirajo tudi spletne seminarje, nekajkrat tedensko pa objavijo zanimive prispevke na svojem blogu, ki je dostopen na `https://cloudant.com/blog/`.

MongoLab MongoLab ima tako imenovano bazo znanja, ki jo najdemo na naslovu `https://support.mongolab.com`. Na tem naslovu najdemo novice in prispevke o pogosto zastavljenih vprašanjih. Uporabnikom je namenjen tudi forum, kjer lahko postavljajo vprašanja v zvezi s svojimi problemi ali pa predlagajo nove funkcionalnosti. Pošljemo jim lahko tudi zahtevo za pomoč, na katero bodo odgovorili v roku enega dne. Naročnikom namenskega načrta, bodo na elektronsko pošto odgovorili hitreje, prav tako so jim na voljo neprestano. Ob kreiranju računa in prvi vzpostavitvi do MongoDB baze, so na nadzorni plošči jasni napotki in dodatne povezave na dokumentacijo MongoDB. Vprašanja se lahko postavljajo tudi na Twitterju na `@mongolab`, skupnost pa se nahaja tudi na socialnih omrežjih Google+ (`https://plus.google.com/101536713548676687538/about`) in Facebook (`https://www.facebook.com/MongoLab`). Imajo tudi svoj blog, dostopen na naslovu `http://blog.mongolab.com/`. Kar nekaj vprašanj in odgovorov v poveazi v MongoLabom se nahaja na spletni strani `http://stackoverflow.com`.

Amazon SimpleDB Podobno kot pri Amazon DynamoDB, tudi SimpleDB nudi dokumentacijo, sicer skromnejšo, ki se deli na vodič za začetnike in vodič za razvijalce, v katerem so reference API. Dokumentacija se nahaja na naslovu `http://aws.amazon.com/documentation/simpledb/`. Pogosto

zastavljena vprašanja pa se nahajajo na naslovu `http://aws.amazon.com/simpledb/faqs/`. Za tehnično podporo veljajo enaki paketi kot za Amazon DynamoDB, na enakih mestih lahko tudi postavljamo vprašanja in sledimo novicam.

Dydra Dydra ponuja skromno dokumentacijo na naslovu `http://docs.dydra.com/`, ki vsebuje kratek vodič za začetnike, pogosto zastavljena vprašanja in REST API. V primeru dodatnih vprašanj lahko uporabniki skupnost najdejo na skupini Google (`https://groups.google.com/forum/#!forum/dydra`, IRC kanalu (`#dydra`), GitHub-u (`https://github.com/dydra`) in Twitterju na `@dydra`. Ker pa je storitev v beta različici in ima dostop le malo ljudi, je skupnost zaenkrat še zelo majhna. Zato pa so na voljo preko elektronske pošte in na Skypu.

Sklep Najslabšo podporo uporabnikom in skupnosti ima Amazon SimpleDB, saj gre za starejšo tehnologijo, ki jo je veliko ljudi zamenjalo za Amazon DynamoDB. Zato je dokumentacija pri AmazonDynamoDB bolj obsežna in ažurna. Vendar Amazon DynamoDB nima brezplačne tehnične podpore, zato je navaden uporabnik prepuščen prebiranju dokumentacije in spletnih strani. MongoLab, Cloudant in Dydra nudijo veliko načinov ažurnega komuniciranja z uporabniki, kjer je odzivni čas res izjemen. Pri Cloudantu in Dydri velja izpostaviti tudi dejstvo, da so strokovnjaki na voljo tudi preko telefona ali Skypa, kar nakazuje na oseben odnos pri poslovanju, ne glede na to, koliko denarja zapravi stranka.

4.1.7 Dostop do podatkovne baze

Način kako dostopamo do podatkovne baze je eden izmed pomembnejših kriterijev. Ker je vsak uporabnik navajen na drugačne programske jezike in razvojna okolja, je najboljše, da ponudnik poskusi omogočiti čim več različnih dostopov. Eden izmed dostopov je vmesnik API, ki je pomemben del vsakega ponudnika nerelacijskih podatkovnih baz, saj pripomore k enostavnejši

uporabi in funkcionalnosti. Ponudniki podatkovnih baz kot storitev se poslužujejo tudi vgrajenih nadzornih plošč, kjer lahko pregledujemo podatkovne baze in nad njimi izvajamo poizvedbe.

Amazon DynamoDB Do podatkovne baze Amazon DynamoDB lahko dostopamo na tri načine. Prvi način je uporaba Amazon DynamoDB API-ja preko zahteve HTTP POST. Na ta način lahko izvajamo naslednje operacije: `BatchGetItem`, `BatchWriteItem`, `CreateTable`, `DeleteItem`, `DeleteTable`, `DescribeTable`, `GetItem`, `ListTables`, `PutItem`, `Query`, `Scan`, `UpdateItem`, `UpdateTable`. Naslednji način je uporaba AWS SDK-jev (ang. *Software Development Kit*), ki poenostavljajo klice API. Na voljo je mnogo AWS SDK-jev, ki podpirajo: .NET, Java, PHP, Android, iOS, Ruby. Če na primer uporabljamo razvojno okolje Eclipse, lahko prenesemo AWS zbirko orodij (ang. *Toolkit*) za Eclipse, ki vključuje AWS SDK za Javo, skupaj z dokumentacijo, vodičem in primeri za kreiranje aplikacij. Za dostop pa imamo tudi na voljo uporabo nadzorne plošče AWS (ang. *AWS Management Console*), v kateri lahko ustvarjamo, posodabljamo in brišemo tabele brez pisanja kode. Prav tako lahko pregledujemo elemente v tabelah in izvajamo poizvedbe.

Cloudant Cloudant ima RESTful API, ki je zelo podoben API-ju podatkovne baze CouchDB in temelji na HTTP, s katerim lahko dostopamo do vsakega dokumenta JSON v naši podatkovni bazi preko URL-ja. Cloudant API je primarna metoda za dostopanje in spreminjanje podatkov. Cloudant podpira naslednje zahteve HTTP: GET (pridobivanje podatkov), HEAD, POST (dodajanje dokumentov in vrednosti), PUT (kreiranje novih objektov - podatkove baze, dokumenti, pogledi), DELETE (brisanje objektov) in COPY (kopiranje objektov). Ob vsaki zahtevi vrne nazaj odgovor, ki vsebuje statusno kodo HTTP in pripadajoč statusni odgovor. Ker uporablja odprte, dobro dokumentirane standarde za temelj API-ja, brez težav dostopamo do baze v mnogih programskih jezikih (C#, C, Haskell, JavaScript, Python, Ruby, Scala), s pomočjo knjižnic. Do podatkov lahko dostopamo tudi preko nadzorne plošče Cloudant, kjer lahko določimo dovoljenja za posamezno po-

datkovno bazo, brskamo po dokumentih, ustvarjamo nove dokumente in podatkovne baze. Na voljo nam je tudi spletna administrativna konzola Futon, kjer lahko ustvarjamo podatkovne baze in dokumente, trenutno pa še ne moremo izdelovati novih pogledov.

MongoLab Podatkovne baze MongoLab so dostopne na tri načine, in sicer z gonilniki MongoDB, preko mongo lupine in preko MongoLab RESTful API-ja. Za dostop preko gonilnikov lahko izberemo tistega, ki nam najbolj odgovarja. Obstaja namreč mnogo različnih MongoDB gonilnikov: C, C++, C#, Erlang, Java, JavaScript, Node.js, Perl, PHP, Python, Ruby, Scala in še mnogo drugih, ki so napisani in podprti s strani skupnosti. MongoLab API pokriva kar nekaj ukazov (`getLastError`, `getPrevError`, `ping`, `profile`, `repairDatabse`, `resetError`, `whatsmyuri`, `convertToCapped`, `distinct`, `findAndModify`, `geoNear`, `reIndex`, `collStats`, `dbStats`), ki jih lahko najdemo v gonilnikih MongoDB, in jih lahko uporabimo v RESTful vmesniku preko zahtev HTTPS. Preko vmesnika lahko tudi brskamo po dokumentih v zbirkah, znotraj različnih podatkovnih baz. Omogoča tudi kreiranje, spreminjanje in brisanje dokumentov, niso pa še podprte bulk operacije. Ustvarimo lahko tudi standardne in geoprostorske indekse. Podprte so tudi standardne poizvedbe, z izjemo `map/reduce`, ki jih lahko tudi shranimo za kasnejšo uporabo. Za osnovno uporabo zadostuje uporaba RESTful API-ja, za bolj napredne uporabnike, ki želijo vse funkcionalnosti MongoDB, pa priporočamo uporabo gonilnikov MongoDB.

Amazon SimpleDB Do podatkovne baze SimpleDB lahko, podobno kot pri DynamoDB, dostopamo s knjižnicami AWS SDK za naslednje programske jezike: Java, PHP, Python, Ruby, .NET. Vsebina knjižnic je slabša kot pri DynamoDB. Uporabimo lahko tudi javansko knjižnico Typica, API za nekatere Amazonove storitve, med njimi tudi za SimpleDB. Na voljo je SimpleDB API, ki omogoča izvajanje naslednjih operacij: `BatchDeleteAttributes`, `BatchPutAttributes`, `CreateDomain`, `DeleteAttributes`, `DeleteDomain`, `DomainMetadata`, `GetAttributes`, `ListDomains`, `PutAttributes`, `Select`.

Izvajamo lahko zahteve REST, ki vračajo odgovor v obliki XML dokumenta, ki je v skladu s shemo. Dostopamo pa lahko tudi preko vmesnika JPA (ang. *Java Presistence API*), preko zahtev REST API-ja.

Dydra Za dostop do podatkovne baze Dydra lahko uporabimo API ali vmesnik pa spletni strani. Trenutno obstajata dva API-ja, standardna SPARQL končna točka (ang. *endpoint*) in REST API. REST API lahko uporabimo tudi preko ukazne vrstice, s pomočjo gema Ruby, ki lahko izvede skoraj vse API ukaze. REST API podpira naslednje zahteve HTTP: GET, PUT, POST, DELETE, s katerimi lahko izvajamo operacije nad repozitorijem in uporabniškim računom. Na voljo je tudi napredni spletni vmesnik, kjer lahko dodajamo podatke (bodisi naložimo lokalno datoteko bodisi podamo naslov, kjer se nahajajo podatki) in izvajamo poizvedbe. Pogosto uporabljene poizvedbe si lahko shranimo za kasnejšo uporabo, meri se tudi čas izvajanja poizvedbe.

Sklep SimpleDB je najmanj fleksibilen pri dostopu do podatkovne baze, je tudi edini ponudnik brez spletnega vmesnika. Dydra je zopet v slabšem položaju, ker je še v beta različici, zaradi česar še ni napisanih veliko knjižnic za različne dostope, vendar ima dober spletni vmesnik. Najbolj dostopen je MongoLab, zaradi fleksibilnosti podatkovne baze MongoDB, ki ima na voljo veliko različnih gonilnikov in dobro dokumentacijo pa tudi dober spletni vmesnik. Malo slabša spletna vmesnika nudita Cloudant in Amazon DynamoDB.

Poglavje 5

Izgradnja odločitvenega modela

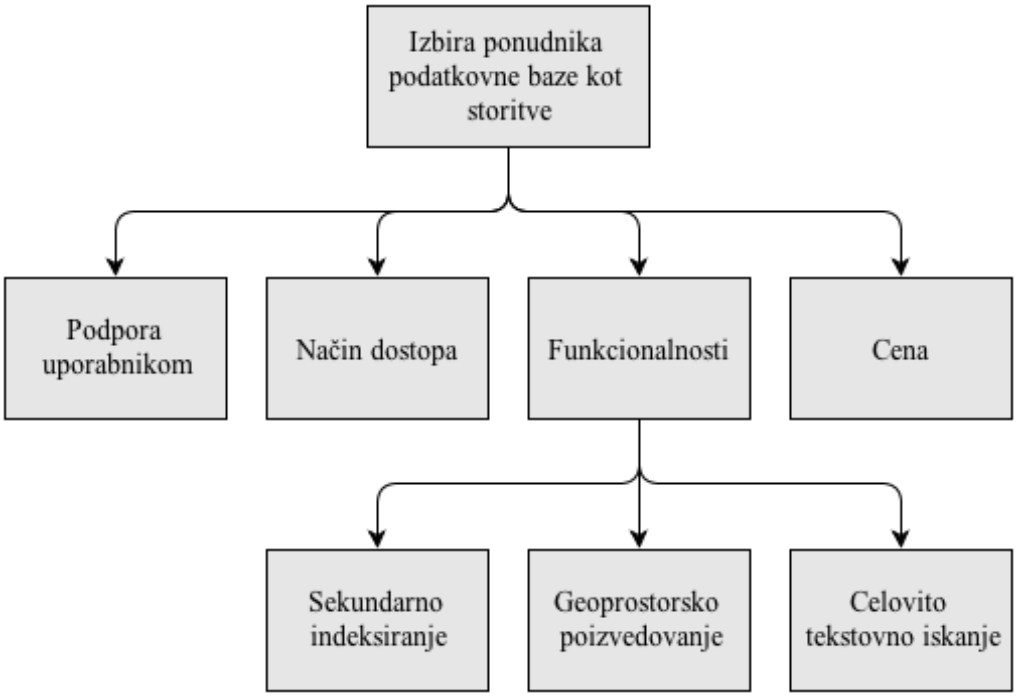
Na podlagi podatkov in sklepov, katere smo predstavili v prejšnem poglavju, smo izdelali odločitveno drevo, v katerem nismo zajeli kategorij podatkovne baze, saj smo že predhodno navedli v katerih primerih uporabe se posamezne kategorije podatkovnih baz najbolje obnesejo. Navedeno se namreč razlikuje, glede na vsak posamičen primer.

Izbrali smo štiri glavne kriterije, ki smo jih upoštevali v odločitvenem drevesu, in sicer: podpora uporabnikom, način dostopa, cena in funkcionalnosti. Pri teh kriterijih je zaloga vrednosti zavzemala tri vrednosti, kar je razvidno v spodnji tabeli.

Podpora uporabnikom	slaba	zadovoljiva	odlična
Način dostopa	slab	zadovoljiv	odličen
Cena	visoka	sprejemljiva	nizka
Funkcionalnosti	slabe	zadovoljive	odlične

Tabela 5.1: Zaloge vrednosti pri glavnih kriterijih.

Kriterij funkcionalnosti smo razdelili na tri podkriterije, ki opisujejo, ali izbrana podatkovna baza podpira določene funkcionalnosti. Te funkcionalnosti so: sekundarno indeksiranje, geoprostorsko poizvedovanje in celovito tekstovno iskanje. Tudi pri teh podkriterijih je zaloga vrednosti obsegala tri vrednosti, kar je razvidno v Tabeli 5.2.



Slika 5.1: Odločitveno drevo.

Sekundarno indeksiranje	ne	da (z omejitvami)	da
Geoprostorsko poizvedovanje	ne	da (z omejitvami)	da
Celovito tekstovno iskanje	ne	da (z omejitvami)	da

Tabela 5.2: Zaloge vrednosti podkriterijev pri kriteriju funkcionalnost.

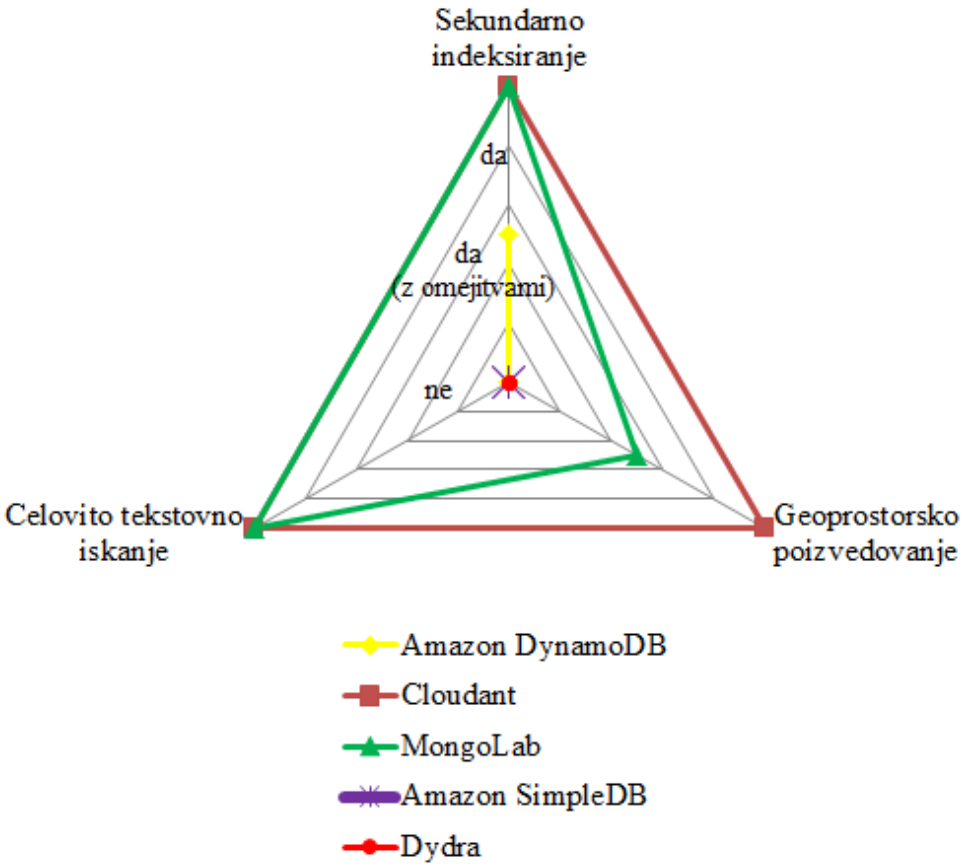
Abstraktne zaloge vrednosti smo ovrednotili na naslednji način:

- Prva vrednost v zalogi vrednosti, torej slab/visoka/ne, je bila ovrednotena z 0 točkami.
- Druga vrednost v zalogi vrednosti, torej zadovoljiv/sprejemljiva/da (z omejitvami), je bila ovrednotena z 0.5 točke.
- Zadnja vrednost v zalogi vrednosti, torej odličen/nizka/da, je bila ovrednotena z 1 točko.

Če je zaloga vrednosti *Funkcionalnost* ovrednotena z nič do ene točke, se preslika v vrednost *slabe*, z eno do vključno dve točki, se preslika v vrednost *zadovoljive* in če je ovrednotena z več kot dvema točkama, se preslika v vrednost *dobre*. Glede na podan kriterij, so funkcionalnosti Cloudanta in MongoLaba dobre, obeh Amazonovih storitev in Dydre pa slabe.

Kot je razvidno iz slike 5.2, je pri kriteriju funkcionalnost najbolje ovrednoten Cloudant, saj podpira vse testirane funkcionalnosti. Za malenkost slabši je ponudnik MongoLab, zaradi omejitve pri geoprostorskem poizvedovanju, saj ne podpira tri dimenzionalnega geoprostorskega indeksa. Na tretjem mestu je ponudnik Amazon DynamoDB, ki omogoča le sekundarno indeksiranje, vendar z omejitvami, saj lahko kreiramo zgolj pet sekundarnih indeksov. Najslabše sta pri kriteriju funkcionalnost ovrednotena Amazon SimpleDB in Dydra, vendar bo v prihodnosti Dydra pridobila nekaj mest, saj trenutno razvijajo geoprostorsko poizvedovanje.

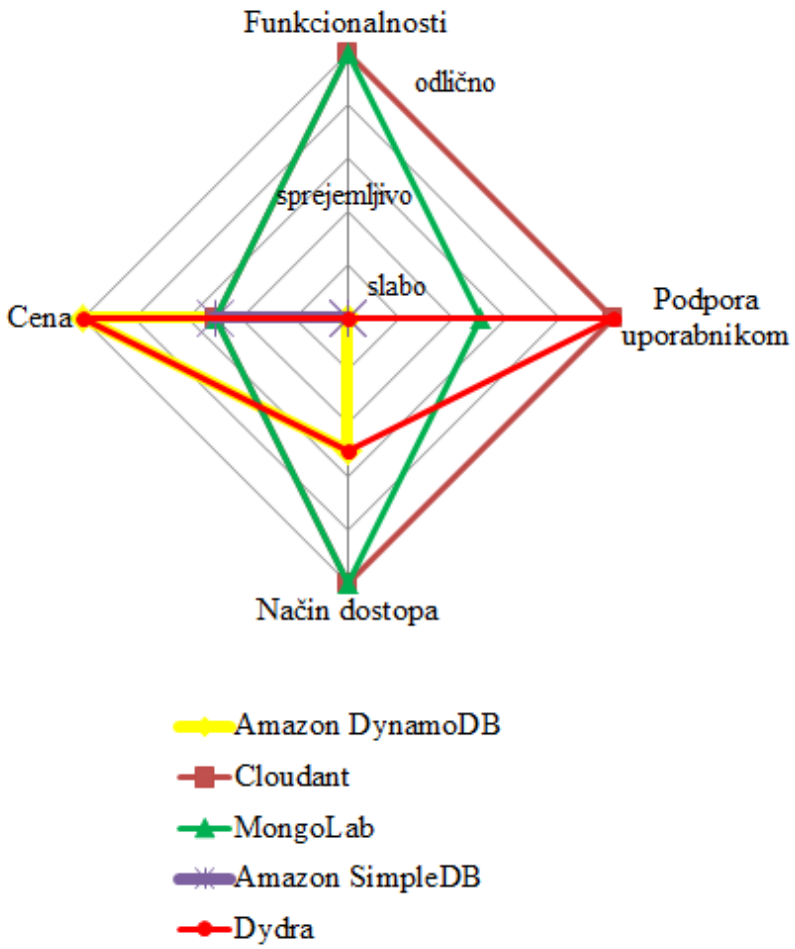
Podobno smo ovrednotili tudi korenski kriterij *Izbira ponudnika podatkovne baze kot storitve*, kjer se vrednosti od nič do vključno ene točke presli-



Slika 5.2: Radarski graf podkriterijev funkcionalnosti.

kajo v vrednost *slab*, vrednosti od ene do treh točk v vrednost *zadovoljiv* ter vrednosti od vključno tri do vključno štiri točke pa v vrednost *odličen*.

Na podoben način smo ocenjevali tudi ponudnike pri glavnih kriterijih. Zopet je na prvem mestu Cloudant, ki se odlično obnese pri kriterijih funkcionalnost, podpora uporabnikom in način dostopa. Ponovno mu sledi MongoLab, ki ima za malenkost slabšo podporo uporabnikom. Pri kriteriju cena sta ocenjena slabše, saj sta dražja od Amazonove storitve DynamoDB. Na tretjem mestu je predstavnik graf podatkovne baze, Dydra, ki si je prislužila odlično oceno pri podpori uporabnikom in ceni, saj je zaenkrat edina popolnoma brezplačna storitev, ne glede na porabo. Na zadnjih dveh mestih sta Amazonovi storitvi, ki ponujata ugodne storitve, vendar najslabšo podporo uporabnikom in podpirata najmanj funkcionalnosti.



Slika 5.3: Radarski graf glavnih kriterijev.

Poglavje 6

Zaključek

Nerelacijske podatkovne baze kot storitve zavzemajo vedno večji delež načina shranjevanja podatkov. Ljudje se odločajo za uporabo podatkovnih baz kot storitev zaradi razbremenitve nekaterih virov, kot so čas, denar in delovna sila. Poleg razbremenitve določenih virov, lahko motive najdemo tudi v potrebi po visoki razpoložljivosti, sposobnosti porazdelitve podatkov med večimi strežniki in hitro obdelavo ogromnih količin podatkov.

V diplomskem delu smo predstavili pet ponudnikov podatkovnih baz kot storitev, ki so povečini izhajali iz različnih kategorij nerelacijskih podatkovih baz, z namenom zajetja čim večih kategorij. Predstavili smo nekaj kriterijev, na podlagi katerih lahko vrednotimo podatkovne baze kot storitve. Kljub temu je pomembno, da pri izbiri ponudnika izhajamo iz našega problema in ponudnika izberemo na podlagi naših zahtev. Navedeno pomeni, da se naš izbor ponudnikov razširi, če naša aplikacija ne potrebuje geoprostorskega poizvedovanja, vendar se lahko ponovno omeji zaradi kakšne druge funkcionalnosti.

Ugotovili smo, da sta najbolj vsestranska ponudnika oba predstavnika dokumentno usmerjenih podatkovnih baz, in sicer Cloudant in MongoLab. Amazon SimpleDB se počasi opušča, saj ga je izrinil njegov naslednik, Amazon DynamoDB. V prihodnosti bo zanimiv razvoj graf podatkovne baze Dydre, saj so ustanovitelji zaenkrat napovedali razvoj geoprostorskega po-

izvedovanja, do končne različice pa bodo dodali še več funkcionalnosti in načinov dostopa.

V tem diplomskem delu nismo obravnavali hitrosti poizvedovanja, saj večina ponudnikov še ni razvila funkcionalnosti za beleženje hitrosti operacij. Pri nekaterih je že v razvoju. Dodajanje novih kriterijev, kot na primer hitrost poizvedovanja, so lahko smernice za nadaljno delo. Diplomsko delo bi sicer lahko poglobili tudi tako, da bi razširili seznam ponudnikov nerelacijskih podatkovnih baz in podrobneje primerjali ponudnike znotraj iste kategorije podatkovnih baz.

Literatura

[1] "DBaaS poised to drive next-generation database growth," September 2013. [Online]. Available: https://451research.com/report-short?entityId=78105&referrer=marketing

[2] Gartner, "IT glossary - cloud computing," Avgust 2013. [Online]. Available: http://www.gartner.com/it-glossary/cloud-computing/

[3] P. Geršič, "Računalništvo v oblaku prihaja tudi nad dolenjsko in belo krajino," Avgust 2013. [Online]. Available: http://www.gzdbk.si/si/aktualno/uspeh/detajl/?id=1200/

[4] S. Bobrowski, "What Exactly Is Database as a Service," Avgust 2013. [Online]. Available: http://dbaas.wordpress.com/2008/05/14/what-exactly-is-database-as-a-service/

[5] O. Corporation, "Database as a Service Reference Architecture - An Overview," Avgust 2013. [Online]. Available: http://www.oracle.com/technetwork/topics/entarch/oes-refarch-dbaas-508111.pdf/

[6] R. Cattell, "Scalable SQL and NoSQL data stores," Avgust 2013. [Online]. Available: http://cattell.net/datastores/Datastores.pdf/

[7] Wikipedia, "No SQL," Avgust 2013. [Online]. Available: http://en.wikipedia.org/wiki/NoSQL

[8] M. Chapple, "Abandoning ACID in Favor of BASE," Avgust 2013. [Online]. Available: http://databases.about.com/od/otherdatabases/a/Abandoning-Acid-In-Favor-Of-Base.htm

[9] Wikipedia, "Cap theorem," Avgust 2013. [Online]. Available: http://en.wikipedia.org/wiki/CAP_theorem

[10] S. Gilbert and N. Lynch, "Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services," Avgust 2013. [Online]. Available: http://lpd.epfl.ch/sgilbert/pubs/BrewersConjecture-SigAct.pdf

[11] S. Tiwari, *Professional NoSQL.* John Wiley & Sons, Inc., Avgust 2011.

[12] C. Strauch, "NoSQL Databases," Avgust 2013. [Online]. Available: http://oak.cs.ucla.edu/cs144/handouts/nosqldbs.pdf

[13] E. Redmond and J. R. Wilson, *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement.* Pragmatic Bookshelf, Maj 2012.

[14] "Riak," September 2013. [Online]. Available: http://basho.com/riak/

[15] "Redis," September 2013. [Online]. Available: http://redis.io/

[16] "Berkeley DB," September 2013. [Online]. Available: http://www.oracle.com/technetwork/products/berkeleydb/overview/index.html

[17] "Hamster DB," September 2013. [Online]. Available: http://hamsterdb.com/

[18] "DynamoDB," September 2013. [Online]. Available: http://aws.amazon.com/dynamodb/

[19] "Project Voldemort," September 2013. [Online]. Available: http://www.project-voldemort.com/voldemort/

[20] "Redis to go," September 2013. [Online]. Available: http://www.redistogo.com

[21] W. Vogels, "Amazon DynamoDB – a Fast and Scalable NoSQL Database Service Designed for Internet Scale Applications," Avgust 2013. [Online]. Available: http://www.allthingsdistributed.com/2012/01/amazon-dynamodb.html/

[22] Amazon, "Amazon DynamoDB Developer Guide," Avgust 2013. [Online]. Available: http://docs.aws.amazon.com/amazondynamodb/latest/-developerguide/Introduction.html

[23] P. J. Sadalage and M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, 1st ed. Addison-Wesley, Avgust 2012.

[24] "MongoDB," September 2013. [Online]. Available: http://www.mongodb.org/

[25] "Apache CouchDB," September 2013. [Online]. Available: http://couchdb.apache.org/

[26] "RavenDB," September 2013. [Online]. Available: http://ravendb.net/

[27] "Terrastore," September 2013. [Online]. Available: https://code.google.com/p/terrastore/

[28] Cloudant, "Cloudant," Avgust 2013. [Online]. Available: https://cloudant.com/

[29] MongoLab, "Mongolab," Avgust 2013. [Online]. Available: https://mongolab.com/welcome/

[30] "MongoHQ," September 2013. [Online]. Available: http://www.mongohq.com/home

[31] "MongoDirector," September 2013. [Online]. Available: http://www.mongodirector.com/

[32] "ObjectRocket," September 2013. [Online]. Available: http://www.objectrocket.com/

[33] "CloudBird," September 2013. [Online]. Available: https://www.cloudbird.net/

[34] "RavenHQ," September 2013. [Online]. Available: http://ravenhq.com/

[35] E. Wolff, "MongoDB and the CAP Theorem," Avgust 2013. [Online]. Available: http://jandiandme.blogspot.com/2013/06/mongodb-and-cap-theorem.html

[36] "Apache Cassandra," September 2013. [Online]. Available: http://cassandra.apache.org/

[37] "Apache HBase," September 2013. [Online]. Available: http://hbase.apache.org/

[38] "Hypertable," September 2013. [Online]. Available: http://hypertable.org/

[39] "Amazon simpleDB," September 2013. [Online]. Available: http://aws.amazon.com/simpledb/

[40] "Instaclustr," September 2013. [Online]. Available: https://www.instaclustr.com/

[41] "Google Cloud Storage," September 2013. [Online]. Available: https://cloud.google.com/products/cloud-storage

[42] Amazon, "Amazon SimpleDB Developer Guide," September 2013. [Online]. Available: http://docs.aws.amazon.com/AmazonSimpleDB/latest/-DeveloperGuide/Introduction.html

[43] J. Tauberer, "Quick Intro to RDF," September 2013. [Online]. Available: http://www.rdfabout.com/quickintro.xpd

[44] Wikipedia, "Resource Description Framework," September 2013. [Online]. Available: http://en.wikipedia.org/wiki/Resource_Description_Framework

[45] "Neo4j," September 2013. [Online]. Available: http://www.neo4j.org/

[46] "HyperGraphDB," September 2013. [Online]. Available: http://www.hypergraphdb.org/index

[47] "InfiniteGraph," September 2013. [Online]. Available: http://www.objectivity.com/infinitegraph

[48] "Apache Jena," September 2013. [Online]. Available: http://jena.apache.org/

[49] "SparkleDB," September 2013. [Online]. Available: http://www.sparkledb.net/

[50] Dydra, "Dydra," September 2013. [Online]. Available: http://dydra.com

[51] W3Schools, "RDF Tutorial," September 2013. [Online]. Available: http://www.w3schools.com/rdf/default.asp

[52] "Electronic Arts," September 2013. [Online]. Available: http://www.ea.com/

[53] "Internet Movie Database," September 2013. [Online]. Available: http://www.imdb.com/g

[54] "SmugMug," September 2013. [Online]. Available: http://www.smugmug.com/

[55] "Overdriver," September 2013. [Online]. Available: http://www.smugmug.com/

[56] "Meteor Solutions," September 2013. [Online]. Available: http://www.meteorsolutions.com/

[57] "Orchestra," September 2013. [Online]. Available: http://www.orchestra.com/

[58] "Cloudant documentation," September 2013. [Online]. Available: https://docs.cloudant.com/

[59] C. Percival, "Dissecting SimpleDB BoxUsage," September 2013. [Online]. Available: http://www.daemonology.net/blog/2008-06-25-dissecting-simpledb-boxusage.html