UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Luka B. Đorović

ANALIZA SLUČAJEVA UPOTREBE RELACIONIH I KOLONSKI ORIJENTISANIH NERELACIONIH BAZA PODATAKA

master rad

Beograd, 2024.

Mentor:

dr Saša Malkov, vandredni profesor
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Ana Anić, vanredni profesor
University of Disneyland, Nedođija

dr Laza Lazić, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: 15. januar 2016.

Ovaj rad posvećujem ...

Naslov master rada: Analiza slučajeva upotrebe relacionih i kolonski orijentisanih nerelacionih baza podataka

Rezime:

Ključne reči: analiza, geometrija, algebra, logika, računarstvo, astronomija

Sadržaj

Glava 1

Uvod

Podaci su najstabilniji deo svakog sistema. Oni su reprezentacija cinjenica, koncepata i instrukcija u formalizovanom stanju spremnom za dalju interakciju, interpretaciju ili obradu od strane korisnika ili mašine. Iako kroz svoju istoriju racunatstvo vazi za oblast koja uvodi nove tehnologije i alate neverovatnom brzinom to nije slucaj za svaku njenu granu. Postoje koncepti koji se kroz istoriju nisu menjali, ili su se slabo menjali i prosirivali. Primera za to ima puno, a to su uglavnom neki univerzanli funkcionalni principi koji se prozimaju kroz racunarske masine, kompilatore, operativne sisteme, sisteme za upravljanje podacima itd. Kada je rec o istoriji sistema za upravljanje podacima, ona se moze podeliti na 3 faze. Na period pre 1970. i Codd-ovog clanka u kojem govori o konceptima Relacionog modelovanja podataka. Zatim na period od 1970 do ranih 2000ih i neprikosnovene vladavine relacionih sistema. Treca faza bi bila period nakon ranih 2000ih kada je doslo do razvoja novih tehnologija pod grupnim imenom „NoSQL" koje se fokusiraju na poznate probleme standardnih relacionih sistema. Kao sto se da zakljuciti, vrlo rano u istoriji racunarstva naislo se na potrebu za standardizacijom mehanizama za obradu podataka.

Pre 70ih rukovanje podacima svodilo se na cuvanje podataka na fajl sistemu operativnog sistema. Apolo sletanje na mesec desilo se u vreme kada nije postojao sistem koji rukuje vecom kolicinopm podataka, sto dodatno govori o velicini takovg poduhvata. Nakon 70ih na talasu Codd-ovog clanka kao i projekta „Sistem R" kao dokaz koncepta da je relacioni model o kojem je Codd pisao moguce implementirati, skoro svaki sistem sa trajnim cuvanjem podataka koristio je relacioni model. Kao lideri komercijalnih proizvoda ovog tipa nametnuli su se IBM i Oracle sa svojim sistemima za upravljanje relacionih baza podataka.

Ipak XXI vek doveo je znacajne alternative u oblasti cuvanja podataka informacionih sistema. Digitalizacija, a samim tim i potreba za obradom vece kolicine podataka, podstakla je nastanak novih tehnologija koje su sluzile da u novonastalom okruzenju omoguce da informacioni sistemi mogu da odgovore na zahteve modernog doba. Problemi te prirode obicno se stavljaju pod grupno ime „problemi velikih podataka" (BigData problems).

Iako su pridosle tehnologije ucestvovale u resavanju tih problema, decenije vladavine relacionih sistema za cuvanje podataka ostavile su dubok trag u praksama rada sa podacima, i sa razlogom predstavljaju defakto standard i dan danas. Sistematizovanje ogromne kolicine fizickog prostora na kojem se podaci mogu cuvati i kasnije koristiti, kao i fleksibilnost strukture podataka sa kojima se radi jesu glavni problemi na koje su se fokusirale tehnologije nastale u NoSQL pokretu. To sa sobom nosi umanjenje stabilnosti i oslanjanja na razvijen ekosistem koju neke organizacije usled striktne biznis logike ne mogu priustiti.

Kolonski orijentisane baze podataka su jedna vazna grupa nerelacionih baza, nastale kao plod BigQuery clanka iz 2004. godine. Njihova glavna odlika je da se podaci organizuju tako da srodni podaci treba da budu blizu jedni drugih kako bi se nad njima mogli primeniti razni optimizacioni algoritmi koji dovode do efikasnijeg skladistenja podataka.

Iz navedenog se naslucuje da nijedan od navedenih koncepata ne prednjaci po defaultu, zato je bitno postojanje materijala koji se bave analizom slucajeva upotrebe navedenih tehnologija. Pored teorijske analize koja se moze pronaci u relevantnim javnim dokumentacijama korisno je imati i konkretne implementacije benchmark-a ciji se rezultati mogu koristiti da se na osnovu njih povuku paralele sa potrebama konkretnih realnih sistema.

Cilj rada .... iz prijave teme

[**?**]-

Glava 2

Modeli za upravljanje podacima

2.1 Relacioni model

Opšte karakteristike

Relacioni model je najpopularniji model za rad sa podacima. On podatke kao i veze izmedju njih predstavlja kroz skup relacija. Iako kao fundamentalna ideja iza relacionog modela stoji tabelarni prikaz podataka, sto uvecava njegovu intuitivnost, korisno je imati na umu formalnu terminologiju koja se koristi u ovakvim sistemima. Svaki red tabele se naziva n-torka. Svaka kolona tabele se zove atribut. Presek reda i kolone je vrednosna celija.

Cesto se javlja dilema oko razlike izmedju tabele i relacije. Tabela je siri pojam, a da bi jedna tabela ujedno bila relacija mora ispuniti sledece uslove: presek kolone i vrste mora predstavljati jedinstvenu vrednost (datum), Sve vrednosne celije jedne kolone pripadaju istom tipu, Svaka kolona ima jedinostveno ime, ne postoje dva identicna reda jedne tabele.

S obzirom da u okviru jedne tabele ne mogu postojati dva identicna reda, jasno je da je pogodno imati nametnutu proceduru koja ne dozvoljava takvu pojavu. U slucaju relacionih modela to predstavlja superkljuc tabele. Superkljuc tabele je kolona ili skup kolona za koje se garantuje da ne mogu uzimati identicne vrednosti za vise redova jedne tabele. Minimalni skup kolona koji predstavlja superkljuc naziva se kljuc kandidat. Svaka tabela ima barem jedan kljuc kandidat za koji nijedna vrednost ne moze biti nepostojeca i taj kljuc kandidat se naziva primarni kljuc. Strani kljuc je kolona ili skup kolona cije vrednosti predstavljaju referencu na odredjeni red neke druge tabele. On igra veliku ulogu u ocuvanju integriteta

baze podataka o cemu ce biti reci u nastavku.

Integritet relacionog modela

Cuvanje integritea relacionog modela predstavlja cuvanje preciznosti i tacnosti podataka koji se cuvaju u bazi. Ono nudi mehanizme ocuvanja konzistentnosti podataka prilikom invazivnih operacija kao sto su dodavanje reda, izmena reda ili brisanje reda u tabeli. Postoji vise vrsta integriteta u relacionom modelu: integritet entiteta, integritet domena, integritet neposojece vrednosti i referencijalni integritet. Integritet entiteta nalaze da se u tabelu ne moze uneti red koji kao primarni kljuc ima nepostojecu vrednost. Integritet domena namece shemu po kojoj svaka kolona moze uzimati vrednost iz unapred dodeljenih skupova vrednosti. Integritet nepostojece vrednosti se govori o eventulnim kolonama cije vretnosti ne mogu kao vrednost imatu nepostojecu vrednost kako se ne bi narusila uspostavljena biznis logika. S bozirom da su asocijacije izmedju relacija determinisane postojanjem ranije pomenutih strani kljuceva u okviru tabele, oni igraju bitnu ulogu u ocuvanju referencijalnog integriteta modela. Referencijalni integritet nalaze da se svaki strani kljuc jedne tabele mora poklapati sa nekim od primarnih kljuceva uparene relacije ili u nekim slucajevima kao vrednost ima nepostojecu vrednost.

Normalizacija

Normalizacja predstavlja jasno definisan proces odlucivanja o tome koji atribut iu relaciji treba da budu grupisani kako bi se izbegla pojava redundantnih podataka. Redundanti podaci zauzimaju prostor na disku i otezavaju odrzavanje sistema. Normalizacija je unapredjivanje logickog dizajna sistema tako da umanjuje dupliranje podataka kao i postizanje inkonzistentosti kroz invazivne operacije nad podacima, ali ne po cenu ocuvanja integriteta baze. Teorija o normalizaciji se zasniva na konceptima normalnih formi. Odredjenoj relaciji se dodeljuje odredjena normalna forma ukoliko zadovoljava pravila vezana za tu normalnu formu. Trenutno postoji 5 definisanih normalnih formi.

ACID

SQL

PostgreSQL

2.2 Kolonski-orijentisani model

Glavne razlike u odnosu na relacioni model

Koncept kolonski orijetnisanih modela svodi se na cuvanje podataka „kolona-ma". To bi znacilo da su sve vrednosti kolone jedne tabele smestene fizicki blizu na disku kako bi se postupak skladistenja mogao optimizovati. Vazna razlika je i ta sto ovakav model nudi fleksibilnost sheme za skladistenje podataka. Ne postoji nista nalik Integritetu domena koji spominjan u 2.1.2. Prednost toga je sto eventualna promena strukture podataka nece bitno uticati na unapred definisanu shemu, kao ni iziskivati migraciju podataka, kao sto bi to bio slucaj kod relacionog modela. Osim toga fleksibilnost sheme se ogleda i u tome sto je broj kolona jednog vektora neogranicen, sto moze biti korisno kod cuvanja nekih agregiranih vrednosti. Posledica fleksibilnosti je nepostojanje nepostojece vrednosti (null). Svaki red koji za neku odredjenu kolonu nema vrednost, nece imati ni informaciju o postojanju te kolone za taj red, pa samim tim nema potrebe ni za cuvanjem bilo kakve vrednosti za tu kolonu. Fleksibilnost sheme sa druge strane utice na nedostatak nametnutog integriteta pa kolonski orijentisan model nije ACID, samim tim nije pogodan za sisteme koji prioritiziraju konzistentost iznad skalabilnosti i performansi.

Optimizacije

Enkodiranje zasnovano na recniku

Enkodiranje po broju ponavljanja

Delta enkoding

BASE

HBase

Glava 3

Slučajevi upotrebe

3.1 Opis i sadržaj eksperimenta

Analiza i uporedjivanje slucajeva upotrebe bice realizovani na osnovu teorijskih i prakticnih izvora i istrazivanja. Svaki primer ce biti pracen eksperimentom koji ce se sastojati od izvrsavanja razlicitih vrsta postupaka. Kao platforma za realizaciju eksperimenata koriscen je host sa docker engine-om. Specifikacije Host-a data je na slici. Pokretanje svakog od eksperimenata je identicno. U okviru repozitorijuma nalazi se sav shell i java kod kao i uputstvo za pokretanje svakog od eksperimenta, zajedno sa deplojment dijagramom.

Svakom eksperimentu dodeljen je precizno definisan kontekst radi uspostavljanja potpune kontrole okruzenja u kojem se eksperiment realizuje. U svrhu definisanja konteksta eksperimenata delom su iskoriscene poznate specifikacije za benchmark bazi podataka. Konkretno za slucaj OLTP okruzenja konsultovana je TPC-C specifikacija, za slucaj OLAP okruzenja konsultovana je TPC-H specifikacija. Distribuirao okruzenje je izuzeto.

Analiza rezultata eksperimenta sprovodi se kroz vise faza. Prva faza je uporedjivanje slozenosti , sto arhitekturalne, sto shematske, realizacije konkretnog slucaja upotrebe kao i da li je konkretan slucaj upotrebe moguce realizovati sa postojecom tehnologijom. Druga faza je uporedjivanje efikasnosti, koja podrazumeva uporedjivanje vremena izvrsavanja programa. Svaka od faza ce ukljucivati tekstualnu diskusiju, slike kao i druge graficke prikaze ukoliko su pogodni.

Kako bi se postigao dovoljan dokaz koncepta (eng. proof of concept), ali i doslednost modernom vremenu, kategorije slucajeva upotrebe koji ce biti obuhvaceni su:

1. Onlajn transakciono procesiranje (OLTP)
2. Onlajn analiticko procesiranje (OLAP)
3. Primena u distribuiranom okruzenju

Kako su za predstavnike izabrani PostgresSQL i HBase, za rezultate merenja iz GLAVE 3 treba uzeti u obzir da implementacija navedenih koncepata nije opsta za sve Relacione sisteme kao ni za sve kolonski orijentisane baze podataka.

3.2 Primena u online transakcionom procesiranju (OLTP)

Online transakciono procesiranje obuhvata kratke, jednostavne, uchestale promene na relativno malom skupu podataka. Primer koji cemo koristiti jeste uopstena transakcija korisnika gde sa jednog racuna treba da se prebaci novac na drugi racun.

Specifikacija PostgresSQL modela:

Specifikacija HBASE modela:

Rezultati:

3.3 Primena u online analitičkom procesiranju (OLAP)

OLAP procesiranje sacinjeno je od skoro iskljucivo citanja podataka. Upiti koji se koriste obicno imaju parametre, imaju visok nivo kompleksnosti i visok procenat podataka kojima pristupa. Primer koji cemo koristiti jeste uopsten primer odrzavanja trgovinskog lanca koji ima skup musterija, proizvoda, dobavljaca, narudzbina. Nas OLAP eksperiment ce se sastojati iz dohvatanja izvestaja o ukupnom kvanitetu, ceni nakon odbijanja poreza, prosecnom popustu za dati status stavke narudzbine.

Specifikacija Postgres modela:

Specifikacija HBASE modela:

Rezultati:

3.4 Primena u distribuiranom okruženju

CAP teorema

Bibliografija

Biografija autora

Vuk Stefanović Karadžić (*Tršić, 26. oktobar/6. novembar 1787. — Beč, 7. februar 1864.*) bio je srpski filolog, reformator srpskog jezika, sakupljač narodnih umotvorina i pisac prvog rečnika srpskog jezika. Vuk je najznačajnija ličnost srpske književnosti prve polovine XIX veka. Stekao je i nekoliko počasnih doktorata. Učestvovao je u Prvom srpskom ustanku kao pisar i činovnik u Negotinskoj krajini, a nakon sloma ustanka preselio se u Beč, 1813. godine. Tu je upoznao Jerneja Kopitara, cenzora slovenskih knjiga, na čiji je podsticaj krenuo u prikupljanje srpskih narodnih pesama, reformu ćirilice i borbu za uvođenje narodnog jezika u srpsku književnost. Vukovim reformama u srpski jezik je uveden fonetski pravopis, a srpski jezik je potisnuo slavenosrpski jezik koji je u to vreme bio jezik obrazovanih ljudi. Tako se kao najvažnije godine Vukove reforme ističu 1818., 1836., 1839., 1847. i 1852.