Cassandra - Wide-Column DB

NoSQL baze podataka

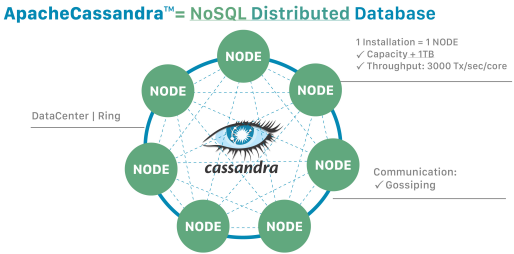
Univerzitet u Novom Sadu
Fakultet tehničkih nauka

Cassandra

- *Zvanična dokumentacija*
- *Go driver*

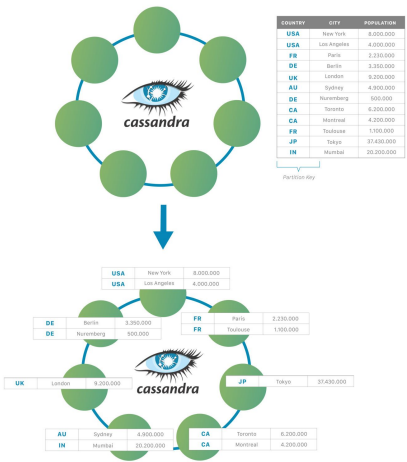
Arhitektura

- **Čvor** - jedan Cassandra server
- **Prsten** - klaster Cassandra servera = *DataCenter*
- **Peer-to-peer** komunikacija između čvorova kroz *Gossiping*



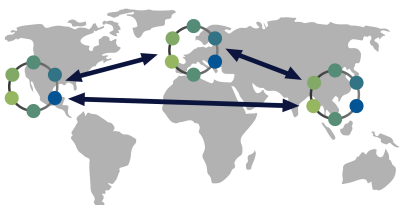
Particionisanje

- Svaki čvor skladišti određeni deo podataka - **particiju**
- **Partition key column** - određuje u kojoj particiji će podatak biti zapisan
- Nije dovoljno čuvati podatke samo u jednom čvoru
  - U slučaju otkaza samo jednog čvora bismo izgubili podatke
  - Loš **fault tolerance**
  - Rešenje?



Replikacija

- **Replikacija podataka**
- U okviru različitih čvorova u prstenu čuvamo duplirane podatke
- **RF** - Replication Factor = diktira broj kopija podataka u okviru baze
- Replike podataka možemo čuvati blizu korisnika koji ih potražuju
- Ali i na udaljenim lokacijama kako bismo osigurali integritet podataka u slučaju otkaza

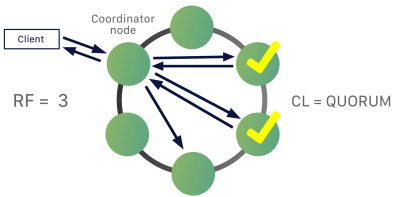


Konzistentnost

- Kako čuvamo podatke na više lokacija, moramo nekako očuvati njihov integritet
- Očuvanje konzistentnosti između više čvorova (i prstenova)
- **Eventual consistency** - baza će u nekom trenutku biti u konzistentnom stanju
- **CL** - Consistency Level = minimalan broj čvorova koji moraju da potvrditi operacije čitanja i pisanja kako bi se one smatrale uspešnim
- Podešavanjem CL vrednosti menja se preferencija performansi, dostupnosti i konzistentnosti podataka

Konzistentnost - upis i čitanje

- **Koordinator** - čvor koji obrađuje zahtev
- Ima ulogu da prosledi upit u čvorove koji skladište odgovarajuću particiju
- Tek kada odgovarajući broj čvorova odgovori, operacija se smatra uspešnom



Tipovi podataka

- Svi osnovni tipovi
  - text (varchar), int, double, date, timestamp, duration
  - uuid, *timeuuid* = uuid + timestamp
- Kolekcije
  - **Mapa** - sortirani set parova ključ-vrednost
    - Vrednost mora biti osnovnog tipa!
    - Moguće je postaviti **TTL** za parove u okviru mape
  - **Set** - sortirana kolekcija jedinstvenih vrednosti; slično mapi
  - **Lista** - sortirana kolekcija vrednosti; slično mapi i setu
- **Tuple** - par vrednosti koje mogu biti različitog tipa
- **UDT** - User Defined Type
  - Mogu se ugnjždavati
  - Mogu sadržati kolekcije
  - **frozen** - immutable; ne mogu se menjati vrednosti pojedinačnih polja

Modelovanje baze podataka - osnovni pojmovi

- **Keyspace** - objekat najvišeg nivoa; definiše RF podataka
- **Tabela** - sadrži redove i kolone sa podacima
- **Red** - torka
- **Partition key column** = PK - određuje u kojoj particiji će podatak biti zapisan, **obavezan!**
- **Clustering key column** = CK + PK = **jedinstveno obeležje reda**; sortiranje i grupisanje
- **Static column** - kolona koja ima konstantnu vrednost na nivou particije
  - Može se koristiti za modelovanje **1-N veza**
- Pristup podacima na osnovu vrednosti drugih kolona sem PK i CK se ne praktikuje i uobičajeno nije moguće!

Pravila dobrog particionisanja podataka

- **Čuvati zajedno ono što se dobavlja zajedno**
  - Ako dobavljamo korisnike na osnovu grada -> grad je partition key
  - Ako dobavljamo komentare na osnovu snimka -> snimak je partition key
- **Izbegavati prevelike particije**
  - Preko 100 000 redova
  - Preko 100 MB
- **Izbegavati česte pristupe samo jednoj od N particija *(hot partition)***
- **Kreirati ograničene particije**
  - **Bucketing** - deljenje particija na manje kroz proširenje partition ključa na više kolona
  - Čuvamo podatke koje prikupljaju IoT senzori i PK nam je id senzora
  - Senzor prikuplja hiljade podataka dnevno -> imaćemo problem velike particije
  - Razdelimo podatke u "kofe" po mesecu kada je očitan podatak za senzor

Proces modelovanje baze podataka

1. Definisanje domenskog modela
2. Definisanje slučajeva korišćenja aplikacije i upita nad podacima
3. Logičko modelovanje baze
4. Fizičko modelovanje baze
5. Evaluacija i unapređenje modela

Najčešće se jedan upit mapira na tačno jednu tabelu!

Primer

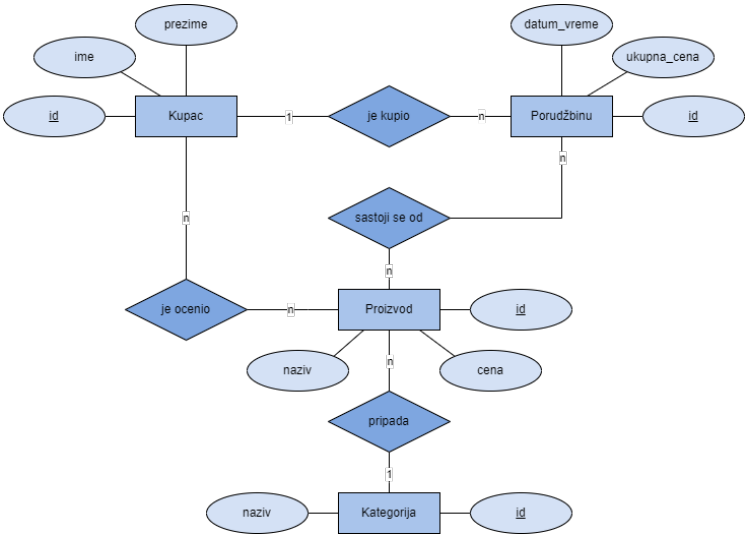
U okviru primera *RestCassandra* upotrebljeni su:

1. *Cassandra* - baza podataka
2. *CQL skripta* - inicijalizacija baze podataka
3. *Docker* - kontejnerizacija rešenja (i "instalacija" baza)
4. *Go* - implementacija primera

Modelovanje baze podataka - zadatak 1

- Kreirati logički model
- Definisati model servisa kroz Go strukture oslanjajući se na *Cassandra gocql* biblioteku
- Podržati sledeće slučajeve korišćenja:
  - Kao kupac, želim da pretražujem sve proizvode koji pripadaju određenoj kategoriji u rastućem poretku cene
  - Kao kupac, želim da vidim istoriju svojih porudžbina počevši od najskorije
  - Kao kupac, želim da vidim istoriju svojih ocenjivanja proizvoda
  - Kao kupac, želim da vidim sve ocene za određeni proizvod

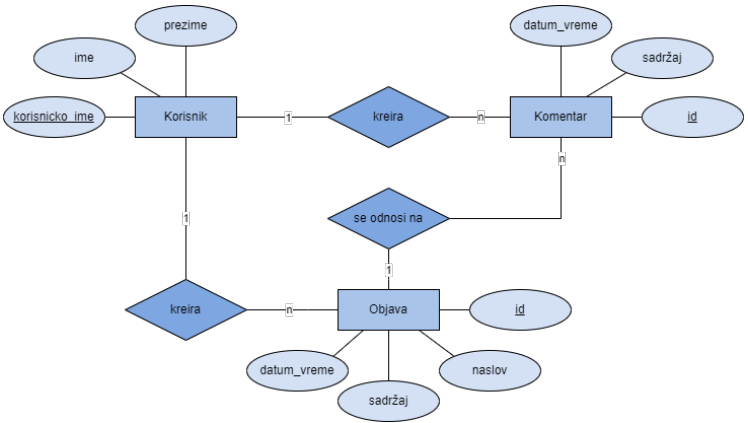
Modelovanje baze podataka - zadatak 1



Modelovanje baze podataka - zadatak 2

- Kreirati logički model
- Definisati model servisa kroz Go strukture oslanjajući se na *Cassandra gocql* biblioteku
- Podržati sledeće slučajeve korišćenja:
  - Kao korisnik, želim da vidim sve svoje objave
  - Kao korisnik, želim da vidim sve komentare vezane za objavu

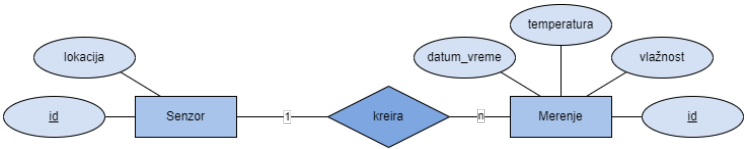
Modelovanje baze podataka - zadatak 2



Modelovanje baze podataka - zadatak 3

- Kreirati logički model
- Definisati model servisa kroz Go strukture oslanjajući se na *Cassandra gocql* biblioteku
- Podržati sledeće slučajeve korišćenja:
  - Kao korisnik, želim da vidim merenja vezana za jednu lokaciju
  - Kao korisnik, želim da vidim merenja vezana za jedan senzor

Modelovanje baze podataka - zadatak 3



Zadaci - bonus

- **Bonus**: Kako bismo proširili sistem da omogući dobavljanje prosečne temperature za određeni vremenski interval?
  - Da li je ovakva baza podataka zgodna za statistike?