**Univerzitet u Nišu**

**Elektronski fakultet**

Replikacija - MongoDB

Predmet: Sistemi za upravljanje bazama podataka

Student: Profesor:

Aleksandar Mitrovic 1444 prof. dr. Aleksandar Stanimirović

# Sadržaj

Replikacija………………………………………………………………………………………………3

Replikacija u MongoDB-u…………………………………………………………………………………5

Proces otkucaja srca…………………………………………..……………………………………8

Odabir set replika…………………………………………………………………………………………15

Sta je Sharding u MongoDB-u…………………………………….………….…………………………17

Replikacija protiv Sharding-a……………………………………………….………………………19

MongoDB set replika protiv MongoDB klastera…………………………….…………………19

Problemi sa kasnjenjem kod replikacije…………………………….………………………19

Konfiguracija replike setova…………………………….………………………19

Literatura……………………………………………………………………………………………………………20

**Replikacija**

Replikacija postoji prvenstveno da bi ponudila redundantnost podataka i visoku dostupnost. Održavamo trajnost podataka tako što čuvamo više kopija ili replika tih podataka na fizički izolovanim serverima. Replikacija je proces kreiranja suvišnih podataka radi pojednostavljenja i zaštite dostupnosti i trajnosti podataka.

Replikacija omogućava povećate dostupnost podataka kreiranjem više kopija podataka na serverima. Ovo je posebno korisno ako server pada ili ako dođe do prekida usluge ili hardverskog kvara.

Ako se podaci nalaze samo u jednoj bazi podataka, bilo koji od ovih događaja bi onemogućio pristup podacima. Ali zahvaljujući replikaciji, aplikacije mogu ostati na mreži u slučaju kvara servera baze podataka, a istovremeno pružaju opcije za oporavak od katastrofe i rezervne kopije.

U nekim slučajevima, replikacija može da obezbedi povećan kapacitet čitanja jer klijenti mogu da čitaju sa različitih servera. Održavanje kopija podataka u različitim centrima podataka može povećati lokalizaciju podataka i dostupnost za distribuirane aplikacije. Takođe može održavati dodatne kopije za namenske svrhe, kao što su oporavak od katastrofe, izveštavanje ili pravljenje rezervnih kopija.

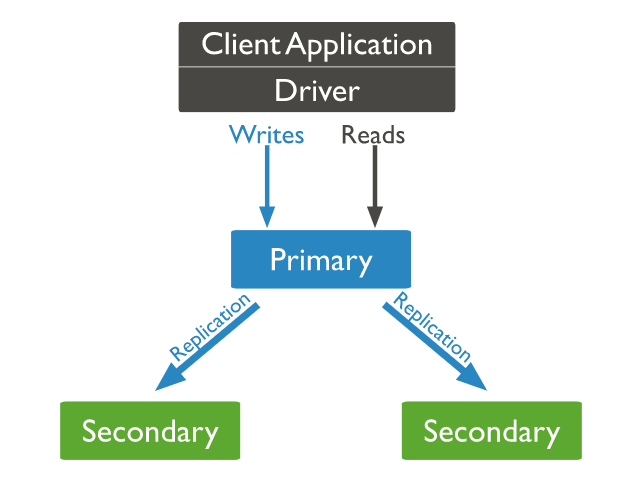
**Replikacija u MongoDB-u**

Sa MongoDB, replikacija se postiže preko skupa replika. Operacije pisanja se šalju na primarni server (čvor), koji primenjuje operacije na sekundarnim serverima, replicirajući podatke.

Skup replika zahteva najmanje tri MongoDB čvora:

• Jedan od čvorova će se smatrati primarnim čvorom koji prima sve operacije pisanja.

• Ostali se smatraju sekundarnim čvorovima. Ovi sekundarni čvorovi će replicirati podatke iz primarnog čvora.



Dok je primarni čvor jedina instanca koja prihvata operacije pisanja, bilo koji drugi čvor unutar skupa replika može prihvatiti operacije čitanja. Oni se mogu konfigurisati preko podržanog MongoDB klijenta.

U slučaju kada primarni čvor nije dostupan ili ne radi, sekundarni čvor će preuzeti ulogu primarnog čvora da obezbedi kontinuiranu dostupnost podataka. U tom slučaju, izbor primarnog čvora se vrši kroz proces koji se zove Izbor skupa replika, gde se kao novi primarni čvor bira najpogodniji sekundarni čvor.

**Proces otkucaja srca**

### Otkucaj srca je proces koji identifikuje trenutni status MongoDB čvora u skupu replika. Tamo čvorovi skupa replika šalju pingove jedan drugom svake dve sekunde (otuda i naziv). Ako bilo koji čvor ne pinguje nazad u roku od 10 sekundi, drugi čvorovi u skupu replika označavaju ga kao nepristupačan.

### Ova funkcionalnost je od vitalnog značaja za proces automatskog prelaska na grešku gde je primarni čvor nedostupan, a sekundarni čvorovi ne dobijaju otkucaje srca u okviru dodeljenog vremenskog okvira. Zatim će MongoDB automatski dodeliti sekundarni server da deluje kao primarni server.

**Odabir set replika**

Izbori u skupovima replika se koriste za određivanje koji MongoDB čvor treba da postane primarni čvor. Ovi izbori se mogu desiti u sledećim slučajevima:

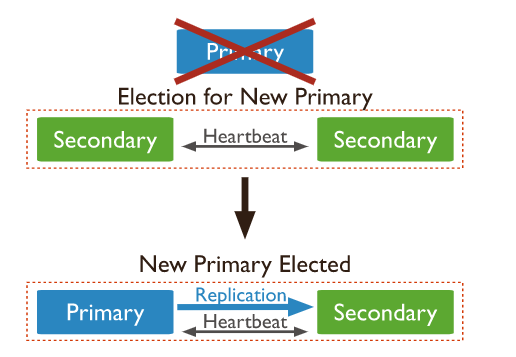
• Gubitak veze sa primarnim čvorom (otkriven otkucajima srca)

• Inicijalizacija skupa replika

• Dodavanje novog čvora postojećem skupu replika

• Održavanje skupa replika korišćenjem metoda **stepDown**  ili **rs.reconfig**

U procesu izbora, prvo, jedan od čvorova će podići zastavicu tražeći izbor, a svi ostali čvorovi će glasati da izaberu taj čvor kao primarni čvor. Prosečno vreme za završetak izbornog procesa je 12 sekundi, pod pretpostavkom da su podešavanja konfiguracije replike u svojim podrazumevanim vrednostima. Glavni faktor koji može uticati na vreme za završetak izbora je kašnjenje mreže i može izazvati kašnjenja u vraćanju vaše replike na rad sa novim primarnim čvorom.

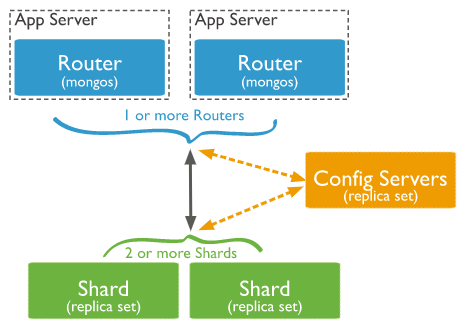


Skup replika ne može obraditi nikakve operacije pisanja dok se izbor ne završi. Međutim, operacije čitanja se mogu izvrsiti ako su upiti za čitanje konfigurisani da se obrađuju na sekundarnim čvorovima.

Srednje vreme pre nego što klaster izabere novu primarnu ne bi trebalo da pređe 12 sekundi, pod pretpostavkom podrazumevanih podešavanja konfiguracije replike. Ovo uključuje vreme potrebno za označavanje primarnih izbora kao nedostupnih i raspisivanje i završetak izbora. Možete da podesite ovaj vremenski period tako što ćete promeniti opciju konfiguracije replikacije **settings.electionTimeoutMillis**. Faktori kao što je kašnjenje mreže mogu produžiti vreme potrebno da se izbori skupa replika završe, što zauzvrat utiče na količinu vremena u kojoj klaster može da radi bez primarnog. Ovi faktori zavise od specifične arhitekture klastera.

Smanjenje opcije konfiguracije replikacije **electionTimeoutMillis** sa podrazumevanih 10000ms (10 sekundi) može dovesti do bržeg otkrivanja primarnog kvara. Međutim, klaster može češće raspisivati izbore zbog faktora kao što je privremeno kašnjenje mreže čak i ako je primarna inače zdrava.

**Sta je Sharding u MongoDB-u?**



MongoDB se izuzetno skalira koristeći tehniku poznatu kao Sharding za rukovanje ogromnim količinama podataka. MongoDB se bavi zahtevima za skladištenje podataka koristeći koncept Sharding, koji uključuje distribuciju podataka i njihovo skladištenje na različitim mašinama.

Sharding koristi tri komponente:

• Shards: To je lokacija na kojoj se čuvaju podaci.

• Config Server: Ovi serveri pomažu u mapiranju podataka iz klastera u Shard-u, koji zatim koriste ruteri upita za obavljanje operacija specifičnih za određeni Shard.

• Query Server: Ovi serveri omogućavaju korisnicima da pristupe i izvršavaju operacije na željenim MongoDB Shardovima.

Postoje dve metode za rešavanje rasta sistema: vertikalno i horizontalno skaliranje.

**Vertikalno skaliranje** podrazumeva povećanje kapaciteta jednog servera, kao što je korišćenje moćnijeg CPU-a, dodavanje više RAM-a ili povećanje količine prostora za skladištenje. Ograničenja u dostupnoj tehnologiji mogu ograničiti da jedna mašina bude dovoljno moćna za dato radno opterećenje.

**Horizontalno skaliranje** uključuje podelu sistemskog skupa podataka i opterećenja na više servera, dodavanjem dodatnih servera radi povećanja kapaciteta po potrebi. Iako ukupna brzina ili kapacitet jedne mašine možda nisu visoki, svaka mašina se nosi sa podskupom ukupnog radnog opterećenja, potencijalno pružajući bolju efikasnost od jednog servera velike brzine i velikog kapaciteta.

MongoDB podržava horizontalno skaliranje putem šaranja. i lako se nosi sa opterećenjem čitanja i pisanja.

……..

**Replikacija protiv Sharding-a**

Replikacija se odnosi na praksu kopiranja podataka sa primarnog serverskog čvora na sekundarne serverske čvorove. Povećava dostupnost podataka i promoviše rezervne kopije, u slučaju da seprimarni server pokvari. Kopira podatke na svakom serveru.

Sharding se odnosi na proces rukovanja horizontalnim skaliranjem na različitim serverima pomoću zajedničkog ključa. Kopiraćete podatke holistički tako što ćete podeliti kopije delova podataka u različite skupove replika. Svi ovi skupovi replika rade zajedno kako bi iskoristili sve podatke.

Kada deljenje i replikacija rade zajedno, oni se nazivaju deljeni klaster. Svaki deo se replicira da bi se sačuvala ista dostupnost podataka

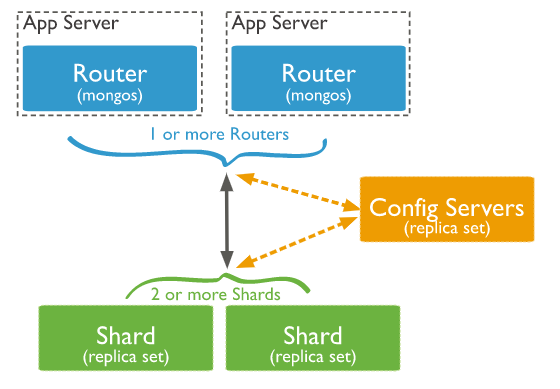
MongoDB podeljeni klaster se sastoji od sledećih komponenti:

• Shard: Svaki deo sadrži podskup podeljenih podataka. Svaki deo se može primeniti kao skup replika.

• mongos: Mongos deluje kao ruter upita, obezbeđujući interfejs između klijentskih aplikacija i podeljenog klastera. Počevši od MongoDB-a 4.4, mongos može da podrži zaštićeno čitanje kako bi se minimizirala kašnjenja.

• konfiguracioni serveri: Konfiguracioni serveri čuvaju metapodatke i podešavanja konfiguracije za klaster.

Sledeći grafikon opisuje interakciju komponenti unutar podeljenog klastera:



**MongoDB set replika protiv MongoDB klastera**

Skupovi replika kreiraju nekoliko kopija istog zapisa kroz čvor skupa replika. Osnovni ciljevi MongoDB skupa replikacije su sledeći:

• Dostupnost podataka

• Odredite ugrađeno rešenje za pravljenje rezervnih kopija

Klasterski rad je drugačiji. MongoDB klaster distribuira podatke kroz više čvorova pomoću ključa shard-a. U ovom procesu podaci se dele na više naslova. One se zovu shard-ovi i kopiraju svaki shard u drugi čvor. Glavna svrha klastera je da podrži veoma velike zapise podataka i operacije velike propusnosti horizontalnim skaliranjem radnih opterećenja. Glavne razlike između skupa replika i klastera su:

• Skup replika kopira ceo skup podataka.

• Klasteri distribuiraju radna opterećenja i skladište podatke (šardove) na više servera.

MongoDB omogućava korisnicima da kreiraju klastere za šaranje kako bi kombinovali ove dve funkcije. U klasteru za dijeljenje, svaki dio se replicira na sekundarni server kako bi se obezbijedila visoka dostupnost podataka i redundantnost.

**Problemi sa kasnjenjem kod replikacije**

Glavna briga kada je u pitanju konfigurisanje replikacije je kašnjenje (lag) replikacije. Ovo se odnosi na kašnjenje u procesu replikacije na sekundarni čvor nakon ažuriranja primarnog čvora u skupu replika.

Određeno kašnjenje u replikaciji tokom replikacije velikih skupova podataka je normalno. Ipak, sledeći faktori mogu povećati kašnjenje replikacije, negirajući prednosti ažurne replikacije:

• Kašnjenje mreže. Pošto imate posla sa višestrukim MongoDB instancama koje se nalaze na različitim serverima tokom replikacije, primarni metod komunikacije će biti mreža. Ako mreža nije dovoljna da zadovolji potrebe procesa replikacije, doći će do kašnjenja u replikaciji podataka u celom skupu replika. Zbog toga je bolje uvek usmeriti svoj saobraćaj u stabilnu mrežu sa dovoljnom propusnošću.

• Propusnost diska. Ako čvorovi replikacije koriste različite tipove diskova (npr. primarni čvor koji koristi SSD, dok sekundarni čvorovi koriste HDD kao diskove), doći će do kašnjenja u replikaciji jer će sekundarni čvorovi sporije obrađivati upite za pisanje u poređenju sa primarnim čvorom. Ovo je uobičajen problem u primeni sa više zakupaca i velikih razmera.

• Velika opterećenja. Izvršavanje teških i dugotrajnih operacija pisanja na primarnom čvoru će takođe dovesti do kašnjenja u procesu replikacije. Dakle, najbolje je da ispravno konfigurišete MongoDB Vrite Concern tako da proces replikacije bude u stanju da prati opterećenje bez uticaja na ukupne performanse skupa replika.

• Pozadinski zadaci. Još jedan važan korak je identifikovanje pozadinskih zadataka kao što su ažuriranja servera, cron poslovi i bezbednosne provere koji mogu imati neočekivane efekte na korišćenje mreže ili diska, uzrokujući kašnjenja u procesu replikacije.

• Operacije baze podataka. – Neki upiti baze podataka mogu biti spori za izvršavanje, dok za neke može biti potrebno dosta vremena. Koristeći profiler baze podataka, možete identifikovati takve upite i pokušati da ih optimizujete u skladu sa tim.

**Konfiguracija replike setova**

Da biste pokrenuli MongoDB instancu, navedite vrednost porta za vašu Mongo instancu zajedno sa putanjom do vaše MongoDB instalacije na vašem sistemu



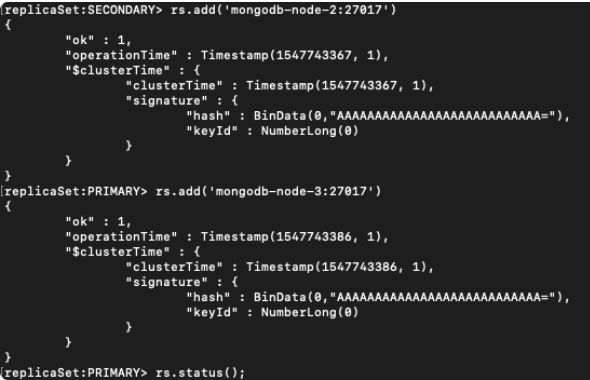
Skup replika sadrži više instanci koje međusobno komuniciraju.

Možete ih povezati koristeći sledeće linije koda:

Kada napravite sve konfiguracije, sada morate da otvorite Mongo Shell sa svojom primarnom instancom i koristite komandu za pokretanje na sledeći način:

Kada inicijalizujete svoj skup replika, sada možete početi da mu dodajete različite MongoDB instance koristeći komandu add na sledeći način:

Na primer, ako želite da dodate node-2 i node-3 u replicaSet1, možete koristiti sledeću komandu u svom Mongo Shell-u:



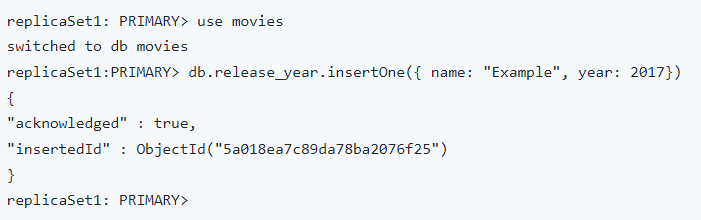
Izlaz {‘ok’:1} označava da je MongoDB instanca uspešno dodata u skup replika. Da biste proverili status replikacije, možete koristiti naredbu status na sledeći način:

Komanda status će proizvesti sledeći izlaz ako su MongoDB instance uspešno dodate:

MongoDB skupovi replika takođe omogućavaju korisnicima da uklone jednu ili više instanci koje su dodali skupu replika koristeći komandu za uklanjanje. Da biste uklonili određenu instancu, prvo morate da je isključite pomoću sledeće komande:

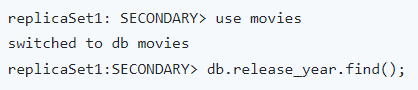
Kada isključite server, potrebno je da se povežete sa svojim primarnim serverom i koristite komandu za uklanjanje na sledeći način:

Možete testirati proces dodavanjem dokumenta u primarni čvor. Ako replikacija radi ispravno, dokument će automatski biti kopiran u sekundarni čvor.

Prvo se povežite sa primarnim čvorom i dodajte dokument koristeći komandu insertOne na sledeći način:

Prebacite se na svoj sekundarni čvor, čvor-2 koristeći sledeću komandu:

Ako je replikacija obavljena ispravno, dokument u vašem primarnom čvoru će automatski biti kopiran u sekundarni. Ovo možete proveriti koristeći komandu find na sledeći način:



Literatura

<https://hevodata.com/learn/mongodb-replication/>

<https://www.mongodb.com/docs/manual/>

<https://www.mongodb.com/basics/>

<https://www.tutorialspoint.com/mongodb>

https://medium.com/condorlabs-engineering/mongodb-replication-what-you-need-to-know-42a62a98b320