Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.41
DOI: https://doi.org/10.24867/02BE07Plavsic

MODELOVANJE DISTRIBUTIVNE ELEKTROENERGETSKE MREŽE UPOTREBOM NEO4J BAZE PODATAKA

MODELING POWER DISTRIBUTION NETWORK USING NEO4J DATABASE

Milica Plavšić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu opisane su graf baze podataka i aplikaciono rešenje koje ima za cilj analizu i pretragu distribuirane elektroenergetske mreže upotrebom Neo4j baze podataka.*
Ključne reči: *Neo4j, Cypher, elektroenergetska mreža*
Abstract – *This work is about graph databases and application solution and its aim was the analyses and search of power distribution network by using Neo4J database.*
Key words: *Neo4j, Cypher, power distribution network.*

1. UVOD

Za ovaj rad su najznačajniji softverski modeli podataka koji su neophodni da bi se proces modelovanja elektroenergetskih sistema obavio uspešno. Naime, elektroenergetski sistem je jedan od najvećih i najuticajnijih upravo zbog toga što se bez njega život ne može zamisliti..Cilj ovog rada je analiza i pretraživanje upravo tih modela, modela elektroenergetske mreže, pri čemu je čuvanje podataka realizovano pomoću Neo4J baze podataka. Osim uvoda i zaključka rad sadrži jos 5 sekcija. U sekciji 2 opisane su relacione i NoSQL baze podataka, a u sekciji 3 su graf orijentisane baze podataka. U sekciji 4 opisana je Neo4j baza podataka i Cypher upitni jezik. Sekcija 5 opisuje CIM model podataka, dok je u sekciji 6 prikazano aplikativno rešenje i prikaz rezultata istog. U poslednjoj sekciji izvedeni su najvažniji zaključci.

2. BAZA PODATAKA

Baza podataka [2] kao deljiva kolekcija međusobno organizovanih i uređenih podataka, danas je najčešće korišćena metoda za čuvanje istih.
Šema baze podataka je apstrakcija na koju se naslanja konkretna baza podataka. Model podataka je apstrakcija koja se koristi za definisanje šeme baze podataka.

2.1. Relacioni model podataka

Relacioni model podataka [2] se zasniva na matematičkom pojmu relacija. Relacija, predstavlja skup n-torki. Relacija je pojava nad šemom relacije. Šema relacija predstavlja apstrakciju odnosno okvir za relaciju. Šema relacije definiše atribute i ograničenja nad njima, koja će zatim relacija (konkretni podaci) poštovati. Relacija predstavlja osnovni tip strukturu gde se podaci čuvaju. Njena tipična reprezentacija jeste tabelarni oblik, gde je svaka vrsta u tabeli jedna torka.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko vanr.prof.

Pomoću torke se svakom obeležju iz nekog skupa obeležja, dodeljuje konkretna vrednost. Svaki podatak je zapisan samo jednom, odnosno u slučaju pravilnog projektovanja, nema redundantnih podataka. Upitni jezik je deklarativan i napredniji od upitnih jezika baza koje nisu relacione. Sistemi za upravljanje bazama podataka su značajno napredniji softveri nego u slučaju nerelacionih baza. Omogućuju bolju validaciju šeme, odnosno forsiranja ograničenja nad podacima. Najnapredniji transakcioni režim nude sistemi za upravljanje relacionim bazama. Velika količina informacija dostupna je na internetu o relacionim bazama.

Skupo je uspostavljanje i održavanje ovakvih baza. Pri projektovanju mora se odrediti količina podata koja može stati u polje da ne dođe do gubitka podataka. U relacionim bazama, da bi se dobavili podaci iz više tabela, potrebno je vršiti spajanje datih tabela (JOIN), što se može vrlo negativno odraziti na performance ovakvih sistema. Nije pogodna za čuvanje nestruktuiranih podataka.

2.2. NoSQL baze podataka

U odnosu na relacione baze podataka, koje podatke predstavljaju putem tabličnih relacija, kod NoSQL baza podataka [1] podaci se smeštaju u različite objekte u zavisnosti od njihove vrste.
NoSQL baze omogućavaju dodavanje podataka bez prethodnog saznanja baze podataka o strukturi podataka koje skladišti. Takođe, NoSQL baze omogućavaju lakši rad na više računara, čime se postiže distribuiranost NoSQL podataka.

Vrste NoSQL baza podataka

Key-value orijentisane: podaci su organizovani u asocijativne nizove ili mape koje koriste jednostavnu metodu ključa i vrednosti za skladištenje podataka.
Dokumentima orijentisane: podaci se čuvaju u dokumentima i smeštaju se u kolekcije. Dokumenti su nezavisni i svaki od njih ima svoj jedinstveni ključ koji služi za dobavljanje podataka.
Colum-family orijentisane: ovaj tip baze predstavlja tabelarnu strukturu u kojoj redovi predstavljaju entitete a različiti redovi mogu sadržati različite kolone i one se mogu dodati nezavisno bilo kom redu u bilo kod trenutku.
I graf orijentisane o kojima će biti reči u narednoj sekciji.

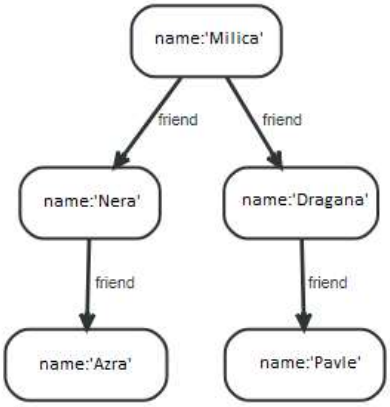
3. GRAF ORIJENTISANE BAZE PODATAKA

Graf baze [1] su kategorija NoSQL baza podataka čija je upotreba sve veća. Graf baze je najadekvatnije koristiti za reprezentaciju i pretraživanje povezanih podataka

značajnijih veličina. Danas se ova vrsta baza veoma često upotrebljava upravo zbog toga što je, zbog prirode podataka, graf najbolji oblik prikaza istih.

3.1. Graf kao model podataka

Graf baze podataka koriste strukturu podataka tipa graf [3] za predstavljanje i skladištenje podatka. Čvor u grafu predstavlja određeni podatak (pojavu entiteta) koji predstavlja objekte u stvarnom svetu. Svaki čvor ima svoj identifikator i svojstva u obliku parova ključ-vrednost, dok više čvorova može imati istu oznaku. Svojstva opisuju čvorove. Veze predstavljaju relaciju odnosno povezanost između dva čvora. Pored toga, sadrže i identifikator, smer veze i početni kao i odredišni čvor.



Slika 1:Primer grafa sa 5 čvorova i 4 veze

Predstavljanje grafa

Uobičajena su dva načina predstavljanja grafa [3]:
Matrica povezanosti - Za graf G=(V,E), neka je $V=\{v_1,v_2 \ldots v_n\}$, matrica povezanosti grafa je kvadratna matrica $A=[a_{ij}]$ reda n, sa elementima $a_{ij}=1$ ako (v_i, v_j) pripada E, a ostali elementi matrice A su nule.
Lista povezanosti - Predstavljanje grafa G=(V,E) pomoću liste povezanosti sastoji se od vektora lista, po jednu za svaki čvor iz skupa V. Za svaki čvor *u* koji pripada V, lista povezanosti $A_{dj}[u]$ sadrži sve čvorove *v* takve da postoji veza (u,v) koja pripada E.

3.2. Modeliranje graf baza podataka

Da bi model grafa bio ispravan i tačan, potrebno je da bude čitljiv i smislen, da baza omogućava lako dodavanje veza i čvorova kao i brisanje postojućih bez narušavanja strukture grafa. Skladište mora biti optimizovano za smeštanje podataka u obliku grafa, a i potrebno je da se mogu izvoditi upiti koji su od značaja i koji daju ispravne i željene rezultate.

3.3. Razlika između graf baze i relacione baze podataka

Kod relacionih baza svaka tabela je kolekcija fiksnih atributa, dok je kod graf baza promena znatno jednostavnija. Vreme izvršavanja postavljenog upita u graf bazi je konstantno, dok u relacionim bazama ono zavisi od broja zapisa u tabeli [4]. Graf baze su optimizovane za veze između podataka dok su relacione primenljivije za agregaciju podataka. Graf baze su adekvatnije za složene strukture podatatak, dok su relacione za relativno jednostavne strukture.

3.4. Prednosti graf baze podataka

Brže se može izvesti upit i to nezavisno od količine podataka. Mogućnost reprezentacije povezanih podataka na prirodan način kao skup objekata (čvorova) povezanih skupom objekata (veza). Model domene može se izraditi u vrlo kratkom roku, pa programer može odmah početi s izradom aplikacije, što ubrzava a i olakšava postupak. Postoji izrazita fleksibilnost na moguće promene tokom razvoja ili nakon isporuke aplikacije.

3.5. Nedostaci graf baze podataka

Niska stopa zrelosti i podrške, usled toga što je ova kategorija relativno nova. Nedostatak podrške za upravljanje većim brojem korisnika zbog čega upravljanje korisnicima mora biti implementirano izravno u aplikaciji. Veći su troškovi implementacije u praksi jer, da bi se sprečila smanjenja u performansama u slučaju više korisnika, graf baze podataka moraju biti implementirane i skladištene na jednom serveru.

4. Neo4j

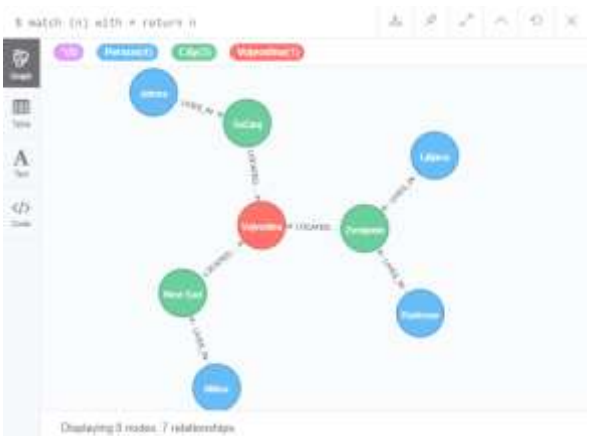
Neo4j [7] je jedna od najpopularnijih baza podataka u okviru graf baza podataka i veoma je jednostavna i za korišćenje a i za implementaciju. Najjednostavnji način korišćenja Neo4j baze podataka jeste preko Neo4j pretraživača za izvršavanje upita koji su napisani u Cypher upitnom jeziku.

4.1. Cypher upitni jezik

Cypher [1] je jednostavan deklarativni upitni jeik za graf baze, koji se ujedno i najčešće koristi. Po strukturi je veoma sličan SQL upitnom jeziku.

Mogućnosti Cypher upitnog jezika

Najznačajnije naredbe za upravljanje podacima u Cypher-u su naredbe za projekciju rezultata, naredbe za čitanje, pod-naredbe za čitanje, naredbe za pisanje, naredbe za čitanje/pisanje.



Slika 2: Primer grafa u Neo4j bazi podataka

5. CIM

CIM [5] je apstraktni model kojim se mogu predstaviti najvažniji entiteti elektroenergetske mreže. Svi objekti iz realnog sistema poput generatora, transformatora, prekidača, vodova i slično predstavljeni su klasama u CIM modelu. CIM predstavlja skup paketa definisanih korišćenjem objektno orijentisanih tehnika modelovanja.

5.1. Osnovne klase CIM modela

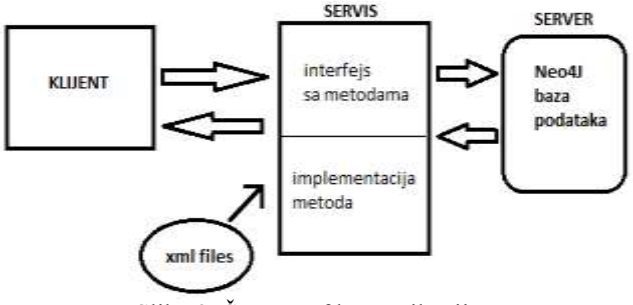
Većina CIM klasa zasniva na *IdentifiedObject* klasi te se ista može uzeti kao osnova i proglasiti glavnom u hijerarhiji. Pored *IdentifiedObject* klase, u okviru elektroenergetskog sistema izdvajaju se i *PowerSystemResource, Equipment, EquipmentContainer, ConductingEquipment, Company* kao i *PSRType.*

6. MODELOVANJE ELEKTROENERGETSKE MREŽE

Bazične komponente CIM modela su transformatorske stanice i fideri odnosno izvodi iz istih. U ovom radu unutrašnjost transformatorskih stanica i fidera biće modelovana u graf bazu podataka. Svrha rada je brzo pretraživanje graf baze podataka za određene upite koji su od značaja. Deo sistema za prenos i distribuciju električne energije predstavlja transformatorsku stanicu. Fideri polaze iz transformatorske stanice. Fideri opisuju opremu i elektroenergetske veze na koje se priključuju potrošači. Fideri sa svojim grananjem čine distributivnu mrežu elektroenergetskog sistema. U obrađivanom modelu postoji 88 transformatorskih stanica i 243 fidera (izvoda) koji su sa svojim karakteristikama izdvojeni u xml fajl [6].

6.1. Realizacija rešenja

Aplikativno rešenje rađeno je u integrisanom razvojnom Visual Studio okruženju. u C# programskom jeziku. Korišćena je Microsoft tehnologija koja se naziva WCF(Windows Communication Foundation)[4].Osnovni delovi WCF modela u ovom aplikativnom rešenju su klijent, servis i server. Klijent preko servisa šalje zahtev za dobijanje određenih podataka i da onda servis komunicira sa bazom koja dobavi željene informacije servisu, koji nakon toga odgovara klijentu (slika 3.).



Slika 3: Šema wcf komunikacije

Potrebno je obezbediti komunikaciju između klijenta i servisa, takođe između servisa i servera. Adresa pristupne tačke na kojoj servis sluša dolazne upite predstavlja se u standarnom url formatu.
Klijent preko krajnje tačke(endpointa) pristupa servisu kako bi prihvatio podatke od servisa. Nakon toga kreira se proxy koji preslikava potpise endpoint operacija kako bi na klijentskoj strani mogao da se pozove taj endpoint, odnosno omogućava klijentu da poziva servis operacije, a da sakrije detalje implementacije od istog. Ugovor (contract) definiše funnkcionalnosti koje pristupna tačka pruža i format poruka koje implementirane metode očekuju.
Ovo rešenje podeljeno je u 3 projekta u visual studio-u, a svaki će biti objašnjem u sledećim pod-sekcijama.

WCFService

Servis će se povezati sa Neo4j bazom podataka koja predstavlja server u ovoj aplikaciji. Povezivanje se vrši preko binarnog protokola. GraphService projektom je predstavljena klasa wcf servisa i naredne klase:
ConnectToNeo4j: u ovoj klasi se vrši konekcija na bazu, a pored konekcije, u ovoj klasi implementirane su i metode koje će se slati graf bazi, za izvršavanje upita u Cypher-u.
LoadXmlToDictionary: najžnačajnija klasa jeste upravo ova, u kojoj se vrši učitavanje xml-ova u određenu kolekciju, a zatim i kreiranje upita za pravljenje grafa u Neo4j bazi podataka na osnovu učitanih podataka. Na osnovu izgleda i strukture xml-a, zaključuje se da su čvorovi u grafu svi elementi koji opisuju jednu distributivnu mrežu (transformatori, prekidači, sekcije, *bay*-evi, signali, pristupne tačke i slično). Suština jeste da se kroz učitan xml fajl prođe samo jednom i tom prilikom se napravi string koji sadrži naredbe za kreiranje čvorova u Cypher upitnom jeziku, kao i da se napuni kolekcija koja će za svaki čvor čuvati gid-ove čvorova koji su povezani na njega.
WCFService: u klasi WCFService kreira se objekat klase ConnectToNeo4j čime se vrši konekcija na bazu, a potom se preko reference objekta pozivaju metode koje su implementirane za slanje upita u Neo4j bazu. Sam servis se implementira kao klasa koja nasleđuje interfejs servisa, i ugovor koji se iz interfejsa definiše.Takođe se definiše pristupna tačka sa svim informacijama potrebnim klijentu da bi bio u mogućnosti da komunicira sa servisom.

GraphInterfaces

Ova klasa predstavlja ugovor i ovde su prikazane deklaracije metoda, koje su dostupne korisniku, a implementirane su u WCFService klasi

GraphClient

Kao što je objašnjeno u prethodnom delu rada, klijent preko endpointa pristupa servisu, što se izvrši u podešavanjima. Proxy omogućava da pozovemo metode koje su dostupne na servisu. Stoga, najpre se mora otvoriti kanal za komunikaciju sa servisom. Takođe se na klijentu jednostavno biraju opcije, odnosno izborom na određeni broj, određen zahtev se šalje dalje servisu, koji prikupi tražene podatke od Neo4j baze i vrati odgovor klijentu, koji dalje biva prikazan na konzoli

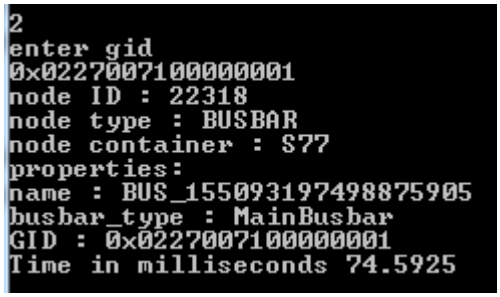
6.2. Analiza rezultata

Prilikom pokretanja aplikacije, potrebno je pokrenuti servis koji će se nakon toga konektovati na bazu i klijenta koji će slati zahteve servisu. Klijent može da izabere jednu od sedam ponuđenih metoda, dok osma predstavlja izlazak iz program. Metode su poređane po rednim brojevima i pored svakog je opisana metoda koja

predstavlja taj određeni broj. U zavisnosti da li klijent želi da upiše neke podatke u bazu, ili da dobavi podatke iz iste, ili pak da izbriše neki određeni čvor, bira određen broj. Izborom na određeni broj servis salje određeni cypher upit ka bazi podataka i javlja se poruka na servisu da je zahtev poslat server. Nakon dobijenog odgovora od servera, servis ga prosleđuje klijentu i na klijentu se prikazuju traženi rezultati na konzoli.

Metode koje su dostupne klijentu su: upis svih transformatorskih stanica i njenih fidera u bazu podataka, pronalaženje čvora sa zadatim gid-om, pronalaženje svih čvorova koji su istog tipa, pronalaženje svih čvorova koji su istog tipa koji pripadaju istom kontejneru, gde se kao parametar prosleđuje ime labele kontejnera, pronalaženje svih čvorova koji su istog tipa koji pripadaju istom kontejneru, gde se kao parametar prosleđuje gid kontejnera, dodavanje novog čvora u bazu i brisanje postojućeg čvora u bazi.

Ako klijent pozove metodu koja kao povratnu vrednost vraća sve atribute o određenom čvoru iz baze podataka nakon pritiska na broj dva, od klijenta se traži da unese gid čvora za koji želi da mu se dobave podaci. Nakon upisanog gid-a rezultati se prikazuju na konzoli klijenta, što se može videti sa slike 4.

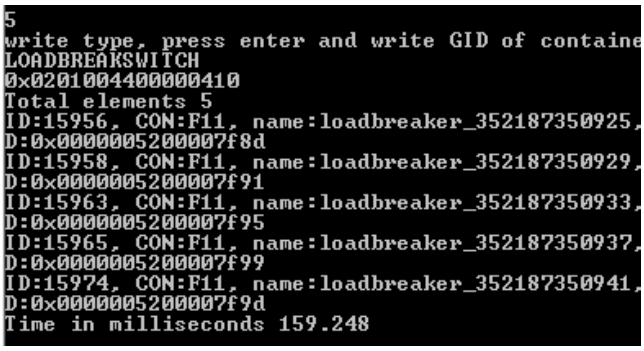


Slika 4:Prikaz rezultata – test1

Takođe se može videti i vreme odziva koje u ovom slučaju iznosi 74.5925ms. Ista metoda pozvana je 20 puta za drugačije zadate gid-ove, a prosečno vreme iznosi 102.48ms.

Prosečno vreme traženja čvora tipa BREAKER pomoću zadate transformatorske stanice pomoću labele, metodom broj četiri je 96.44ms.

Na slici 5. prikazan je rezultat metode koja kao povratnu vrednost vraća sve čvorove koji su određenog tipa, a deo su određenog kontejnera, odnosno određene transformatorske stanice ili fidera.



Slika 5:Prikaz rezultata – test 2

Klijent preko konzole unosi tip čvora, u ovom slučaju je to je LOADBREAKSWITCH, a potom i gid kontejnera, odnosno fider sa gid-om: 0x0201004400000410. Na slici 5. se takođe može videti i tačan broj određenih elemenata u zadatom kontejneru, u ovom slučaju ih je pet. Isto tako se vidi i broj fidera kome pripada čvor, u ovom slučaju je to F11 (fider broj 11), a vreme odziva iznosi 159.248 ms.

Na isti način beležena su vremena odziva za pronalazak čvorova tipa LOADBREAKSWITCH koji pripadaju različitim kontejnerima čiji je gid zadat. Prosečno vreme pretrage iznosi 174 ms.

Vreme odziva metode šest, gde klijent unosi podatke za kreiranje novog čvora u bazu je 158,6737ms. Nakon unošenja novog čvora u bazi, pomoću metode dva koja vraća određeni tip čvora sa zadatim gid-om pronalazi se upravo napravljen čvor za 186.3621ms.

Vreme odziva metode sedam, koja briše čvor sa zadatim gid-om čvora iznosi 249.6253 ms.

7. ZAKLJUČAK

Zadatak ovog rada je bio da se, na osnovu elektroenergetske mreže odnosno transformatorskih stanica i fidera, napravi graf u Neo4J bazi podataka kao i da se izvrše upiti nad tom bazom. Podaci su bili zasnovani na CIM standardu a važno je napomenuti da su bili dati u xml fajlovima. Razvijena je aplikacija koja vrši obradu modela podataka distributivne mreže i kreira upite koji se šalju Neo4J bazi. Na taj način se formira model kome se pristupa od strane razvijenog klijenta u skladu sa specifikacijama za pristup CIM podacima. Dati projekat bi se mogao proširiti u smislu daljih aktivnosti koje bi doprinele da pretraga dostigne veću brzinu. Jedno od rešenja bi moglo biti to da se unapredi algoritam indeksiranjem čvorova u Neo4J bazi jer bi se time ubrzala pretraga grafa a samim tim i pretraga podataka.

LITERATURA

[1] Ian Robinson, Jim Webber, Emil Eifrem: Graph Databases Second Edition, O'Reilly Media, June 2015.

[2] Pavle Mogin, Ivan Luković: Principi baza podataka, Fakultet tehničkih nauka, Novi Sad, 1996.

[3] Milo Tomašević: Algoritmi i strukrute podataka, Akademska misao, 2008.

[4] John Sharp: Windows Communication Foundation 4 Step by Step, Microsoft, 2014.

[5] J.Simmins: Common Information Model Primer, EPRI-Final Report, Palo Alto, California, 2015.

[6] Elliotte Rusty Harold: XML™ Bible, IDG Books Worldwide, 1999.

[7] Adam Fowler: NoSQL For Dummies, John Wiley & Sons, New Jersey, 2015.

Kratka biografija:

Milica Plavšić rođena je u Zrenjaninu 1993. god. Fakultet tehničkih nauka upisala je 2012. god. Diplomski rad iz oblasti Elektrotehnike i računarstva – automatika i upravljanje sistemima, odbranila je 2016. god.