

2) Predlog strategije za particionisanje podataka

2.1) Logovanje u aplikaciju

Od 200 miliona korisnika koji trebaju da koriste aplikaciju, pretpostavlja se da ce vecina njih biti pacijenti. Da bi koristio funkcionalnosti aplikacije, svaki pacijent mora da se uloguje na svoj profil. Zbog velikog broja logovanja, moramo nekako particionisati tabelu USER. Particionisanje ove tabele predlozili bismo kroz horizontalno particionisanje po hes vrednosti kolone ID, koja je random generisana. Razlog ovome je to sto ce bilo koja druga kolona teziti sklonosti, npr. kada bi izabrali prezime, mnogo podataka bi zavrсило u particiji koja pocinje sa M,N,J itd. Stoga ce particionisanje po ID-u mnogo ravnopravnije to uraditi, a predlazemo i hesiranje da se ne bi kojim slucajem, grupisanje podataka u particije desilo i sa ID-em.

2.2)

1)Tabela absence_request

Za tabelu absence_request predlazemo horizontalno particionisanje po koloni status.

2) Tabela complaint

Predlazemo horizontalno particionisanje na osnovu kolone type.

3)Tabela dermatologist_appointment

Predlazemo horizontalno particionisanje na osnovu kolone drugstore_fk, odnosno tome u kojoj apoteci ce se pregled izvorsiti, nakon toga moglo bi se izvorsiti vertikalno particionisanje nad ostalim kolonama.

4)Tabela dermatologist_drugstore

Predlazemo horizontalno particionisanje na osnovu kolone drugstore.

5)Tabela drug

Predlazemo horizontalno particionisanje na osnovu kolone rating.

6)Tabela drug_ingredients

Predlazemo horizontalno particionisanje na osnovu kolone ingredient_id.

7)Tabela drug_order

Predlazemo horizontalno particionisanje na osnovu kolone drugstore.

8)Tabela drug_order_stock

Predlazemo horizontalno particionisanje na osnovu kolone stock_id.

9)Tabela drug_price

Predlazemo horizontalno particionisanje na osnovu kolone drugstore_fk, a zatim i horizontalno particionisanje na osnovu kolone drug_fk.

10)Tabela drug_request

Predlazemo horizontalno particionisanje na osnovu kolone drugstore_id.

11)Tabela drug_reservation

Pretpostavljamo da ce broj pristiglih rezervacija biti preko milion dnevno. Zbog toga predlazemo horizontalno particionisanje na osnovu kolone drug_fk, a zatim horizontalno particionisanje na osnovu kolone drugstore_fk.

12)Table drug_type

Kako ova tabela neće sadržati veliki broj podataka deljenje na particije neće biti potrebno.

13)Tabela drugstock

Predlazemo horizontalno particionisanje na osnovu kolone drugstore_fk, a zatim i horizontalno particionisanje na osnovu kolone drug_fk.

14)Tabela drugstore

Predlazemo horizontalno particionisanje na osnovu kolone rating.

15)Tabela employment

Predlazemo horizontalno particionisanje na osnovu kolone drugstore_id.

16)Tabela ingredient

Kako ova tabela neće sadržati veliki broj podataka deljenje na particije neće biti potrebno.

17)Tabela location

Predlazemo horizontalno particionisanje na osnovu kolone country.

18)Tabela loyalty_configuration

Kako ova tabela neće sadržati veliki broj podataka deljenje na particije neće biti potrebno.

19)Tabela manufacturer

Kako ova tabela neće sadržati veliki broj podataka deljenje na particije neće biti potrebno.

20)Tabela offer

Predlazemo horizontalno particionisanje na osnovu kolone supplier_id.

21)Tabela order_stock

Predlazemo horizontalno particionisanje na osnovu kolone drug_id.

22)Tabela patient_alergens

Predlazemo horizontalno particionisanje na osnovu kolone ingredient_id.

23)Tabela patient_category

Predlazemo horizontalno particionisanje na osnovu kolone name.

24)Tabela patient_category

Predlazemo horizontalno particionisanje na osnovu kolone name.

25)Tabela penalty

Predlazemo horizontalno particionisanje na osnovu kolone pharmacist_fk.

26)Tabela point

Kako ova tabela sadrži dva relevantna podatka koji su geografska širina i dužina ne postoje atributi po kojima bi particionisanje imalo smisla.

27)Tabele rating_dermatologist

Predlazemo horizontalno particionisanje na osnovu kolone rating. A zatim i horizontalno particionisanje na osnovu kolone dermatologist_fk.

28)Tabele rating_drug

Predlazemo horizontalno particionisanje na osnovu kolone rating. A zatim i horizontalno particionisanje na osnovu kolone drug_fk.

29)Tabele rating_drugstore

Predlazemo horizontalno particionisanje na osnovu kolone rating. A zatim i horizontalno particionisanje na osnovu kolone drugstore_fk.

30)Tabele rating_pharmacist

Predlazemo horiznotalno particionisanje na osnovu kolone rating. A zatim i horizontalno particionisanje na osnovu kolone pharmacist_fk.

31)Tabela reply

Predlazemo horiznotalno particionisanje na osnovu kolone system_admin_id.

32)Tabela roles

Kako je broj redova ove tabele veoma mali i neće se menjati nema potrebe za particionisanjem ove tabele.

33)Tabela subscription

Predlazemo horiznotalno particionisanje na osnovu kolone patient_fk.

34)Tabela substitutions

Predlazemo horiznotalno particionisanje na osnovu kolone original_id.

35)Tabela user_role

Predlazemo horiznotalno particionisanje na osnovu kolone role_id.

36)Tabela users

Predlazemo horiznotalno particionisanje na osnovu kolone type.

3.) Replikacija baze i otpornost na greške

Procena je da će aplikacija imati čak 200 miliona korisnika, tako da je definitivno od ključne važnosti bezbednost i pozudanost servera. S obzirom na broj korisnika, novac svakako ne bi trebao da predstavlja problem tako da ćemo ekonomski aspekt ignorisati kada je reč o ovakvim esencijalnim pitanjima. Najpre bi svaka klinika trebala da ima sopstveni server na kojem bi se nalazila lokalna baza te apoteke. Smatramo da bi kopije svih baza mogle da se čuvaju na *blockchain* mreži. Serveri koji održavaju mrežu bili bi raspoređeni po čitavom svetu, na svim kontinentima, i u državama različitog spoljno-političkog opredeljenja, kako bi šansu da se svi serveri fizički ugase od strane nadležnih organa sveli na minimum, i time u velikoj meri doprineli nezavisnosti aplikacije od bilo kojih spoljnih faktora.

4.) Predlog strategije za keširanje podataka

Kao strategija za keširanje podataka, smatramo da bi dobro rešenje bilo keširanje određenog dela podataka kojem se najčešće pristupa, dok bi se ostatku podataka pristupalo

putem baze. Pristup, odnosno pretraga informacije koja nam je potrebna vršila bi se standardno, tako što bi se informacija najpre tražila među keširanim podacima, a ukoliko ne postoji među keširanim podacima, pristupalo bi se bazi i tamo bi se nastavila pretraga datog podatka. Na mesečnom nivou bi se vršila analiza frekventnosti pristupa različitim tipovima podataka, i u skladu sa tim bi se menjali podaci koje čuvamo keširane.

5.) Procena hardverskih resursa potrebnih za skladištenje u narednih 5 godina

U našoj aplikaciji čuvaju se tipovi podataka koji zauzimaju malo memorije kao što su boolean, string, int, double. Predpostavićemo da će na dnevnoj bazi našu aplikaciju koristiti u proseku 1 000 000 korisnika (s obzirom da je reč o aplikaciji za medicinske predloge i rezervaciju lekova, većina korisnika neće učestalo koristiti aplikaciju, već samo po potrebi, par puta godišnje. Ukoliko svako od ovih million korisnika u bazu doda 400 bajtova podataka. Sa ovim pretpostavkama dolazimo do zaključka da će za 5 godina skladištenja podata biti potrebno obezbediti oko 700 gigabajta.

6.) Predlog strategije za postavljenje load balansera

Za algoritam load balansera za koji smatramo da bi radio zadovoljavajućim performansama predlažemo kombinaciju *Least-connection* pristupa i *ip-hash* pristupa. Na taj način bi load balanser zahteve raspoređivao na severe koji u tom trenutku imaju najmanje zahteva, ali bi uzimao u obzir i ip-adresu, i na osnovu lokacije odakle stiže zahtev bi raspoređivao na servere bliže mestu odakle stiže zahtev.

7.) Operacije korisnika koje treba nadgledati u cilju poboljšanja sistema

U nastavku će biti predstavljene operacije za koje smatramo da bi bilo značajno da budu praćene sa ciljem poboljšanja sistema. Uz svaku operaciju će biti naveden i predlog frekventnosti nadgledanja. Operacije koje treba nadgledati su:

- Broj pregleda dermatologa neke apoteke - posmatrao bi se na nivou čitavog sistema, sa ciljem detektovanja apoteke kod kojih je broj pregleda ispod očekivanog nivoa, učinak bi se posmatrao na mesečnom nivou
- Broj pregleda farmaceuta neke apoteke - posmatrao bi se na nivou čitavog sistema, sa ciljem detektovanja apoteke kod kojih je broj pregleda ispod očekivanog nivoa, učinak bi se posmatrao na mesečnom nivou
- Broj prodatih lekova određenog proizvođača u apoteci - posmatrao bi se na nivou čitavog sistema, sa ciljem detektovanja prodaje određenog proizvođača koja je ispod očekivanog nivoa, kako bi se ta apoteka obavestila o slaboj prodaji, i kreirala neku promociju ili akciju sa ciljem poboljšanja prodaje, učinak apoteke bi se posmatrao na mesečnom nivou
- Broj otkazanih pregleda nekog korisnika - posmatrao bi se na nivou apoteke, kako bi se detektovali eventualni spam nalozi, posmatranje bi se vršilo na nedeljnom nivou

- Broj otkazanih rezervacija za lek nekog korisnika - posmatrao bi se na nivou apoteke, kako bi se detektovali eventualni spam nalozi, posmatranje bi se vršilo na nedeljnom nivou
- Broj slabih ocena (manjih od 4) nekog od zaposlenih - posmatrao bi se na nivou apoteke kako bi se uočili radnici koji nisu na potrebnom nivou, posmatranje bi se vršilo na kvartalnom nivou
- Broj novoregistrovanih korisnika neke apoteke - posmatrao bi se na nivou čitavog sistema sa ciljem detektovanja apoteke koje imaju mali broj novoregistrovanih korisnika u odnosu na mesto u kom se nalaze, kako bi se eventualno više novca uložilo u marketing u tim oblastima, posmatranje bi se vršilo na kvartalnom nivou

8.) Dizajn predložene arhitekture

