

Skalabilnost

1. Dizajn seme baze podataka(koceptualni,logicki ili fizicki)

2. Predloziti strategije za partitionisanje podataka

Partitionisanje podataka omogućava laku brzi i laki pristup podacima. Aplikaciju koristi veliki broj korisnika, a samim tim i imamo i previse podataka za cuvanje. Dolazimo do zakljucka da je partitionisanje baze odlicno resenje. Partitionisanje moze da se obavi na nivou samih funkcionalnosti, vremenskom periodu, ulogama.. Na taj nacin bi imali vise razlicitih particija i omogućili brzu i laku komunikaciju. Partitionisanjem po funkcionalnostima bi izdvojili particiju o klijentima(u nasem slucaju pacijentima) i zaposlenima. Ukoliko partitionisemo po vremenskom intervalu, prvo moramo odrediti koji je to vremenski interval. Svaki vremenski interval je zasebna particija.

3. Predlog strategije za replikaciju baze i odredjivanje otpornosti na greske

U nasem projektu je koriscenja PostgreSQL. Ona omogućava replikaciju, odnosno kopiranje promena u klasteru primarne baze podataka u rezervnu bazu podataka na drugom serveru.

Na taj nacin je aplikacija otpornija na greske, jer se prilikom greske sa primarnom(master) bazom rad nastavlja na rezervnoj(slave) bazi.

4. Predlog strategije za kesiranje podataka

Kesiranjem podataka zelimo da odbacujemo informacije koje nece biti potrebne duzi vremenski period. Neki od odlicnih algoritama je CAR (Clock with Adaptive Replacement) koji kombinuje ARC i CLOCK algoritam. Samopodesavajuci je i ne zahteva nikakve parametre od korisnika.

ARC-Adaptive Replacement Cache (keš sa adaptivnom zamenom) upotrebljava informacija o skoro izbačenim podacima iz keša da bi se dinamički određivala veličina zaštićenog i uslovnog segmenta u cilju bolje iskorišćenosti prostora u kešu.

5. Okvirna procena za hardverske resurse potrebne za skladistenje svih podataka u narednih 5 godina.

Vodeci se time da aplikacija ima 200miliona korisnika, a na mesecnom nivou se obavi 1000000 pregleda potreban je veliki broj resursa za skladistenje podataka. Velicina UTF-8 zauzima 4B, na osnovu proracuna koliko karaktera je potrebno za cuvanje pregleda, konsultacija, zalbi, zahteva, anketa, recepata.. Na mesecnom nivou je potrebno oko 450000bytes za skladistenje informacija, sto znaci da u narednih pet godina treba da se obezbedi 30GB!(450000b*12MESECI*5GODINA)

6. Predlog strategije za postavljanje load balansera

Load balanser je odlicna strategija za preventivu otkazivanja nekog od servera, jer otkazivanjem jendog on preusmerava zahteve koji pristizu na dostupan server. Predlazemo round-robin nacin podele opterecenja. Zahtevi ce se redom prosledjivati izlistanim serverima u krug na osnovu redosleda kojim su upisani u upstream backend sekciju.

7. Predlog koje operacije korisnika treba nadgledati u cilju poboljsanja sistema

Trebalo bi imati event-e koji ce pratiti odredjene operacije korisnika, a na taj nacin izvuci statistiku koja ce pomoci u poboljsanju samog sistema. Odlicno bi bilo pratiti broj pregleda po danima, broj rezervacija po danima, broj zalbi po danima, broj registracija po danima, broj konsultacija po danima,

broj otkazivanja pregleda po danima, ocenjivanje lekara, farmaceuta, apoteke, najcesce porucivani lekovi, koliko j vremena potrebno korisniku da zakaze pregled ili izvrsi bilo koju operaciju.

8. Kompletan crtez dizajna predlozene arhitekture(aplikativni serveri, serveri baza, serveri za kesiranje..)

