

Slika 2 Particionisanje baze

Postoje dva osnovna oblika particionisanja podataka:

1. Horizontalno - zasnovano na predikciji vrednosti atributa gde se zapisi postavljaju u particiju zavisno od jednog ili više atributa
2. Vertikalno - relacija se deli tako da se jedan atribut postavlja u jednu particiju, a drugi u drugu, kako bi se omogućilo postavljanje polja kojima se često pristupa na brze uređaje

U našem slučaju, vertikalno particionisanje bi imalo smisla ukoliko bi se model proširio dodatnim poljima. S druge strane, horizontalno particionisanje bi se moglo primeniti na rad sa radnim kalendarom.

3. *Replikacija baze podataka*

Replikacija je alat koji se koristi za distribuciju podataka u druge baze podataka, odnosno obezbeđuje da na više servera postoje iste informacije. U sistemima koji koristi mnogo korisnika, distribuiranje jeste metod koji obezbeđuje visoku funkcionalnost svakom korisniku.

U našem slučaju je pogodno primeniti replikaciju baze upravo iz razloga što veliki broj korisnika sa različitih lokacija pristupa i rukuje velikom količinom podataka. Ukoliko pretpostavimo da naš sistem koristi stanovništvo bivše Jugoslavije, došli smo do najpovoljnijeg rešenja: svaka država bi mogla imati svoju bazu podataka koja bi bila ista kao baze u ostalim zemljama. One bi međusobno komunicirale, međutim, podaci koji se tiču samo jedne države čuvali bi se samo u njenoj bazi. Na taj način bismo ograničili broj korisnika kao i količinu podataka koja se čuva u jednoj bazi. Povećala bi se dostupnost i sigurnost čuvanih podataka.

Takođe, ovakva strategija bi se mogla primeniti i na pokrajine jedne iste države, u zavisnosti od njene geografske veličine kao i veličine stanovništva (potencijalnih korisnika).

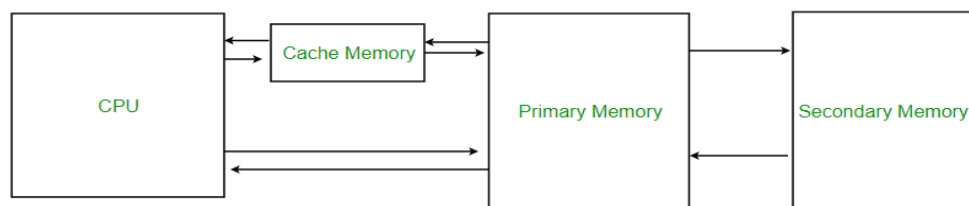
Pored toga, koristeći PostgreSQL bazu podataka, uzeli smo u obzir to da ona između ostalog nudi master-slave režim rada. Tačnije, postoje master i slave baze podataka. Master baza je glavna baza u kojoj se čuvaju svi podaci. Slave baza je backup (rezervna) baza koja se nalazi na standby serveru. Sve transakcije koje se obavljaju na master bazi su uspešno commit-ovane tek ako su uspešno upisane i u master i u slave bazu. Uzimajući sve navedeno u obzir, dolazimo do zaključka da su master i slave baza sinhronizovane. Upravo iz tog razloga, moguće je rasteretiti master bazu i koristiti je samo za upis podataka, dok se jedna ili više slave baza mogu koristiti isključivo za čitanje podataka. Takođe, ukoliko master baza otkáže, slave baza automatski može postati master što nam pruža dodatni stepen pouzdanosti i sigurnosti. Naveden način replikacije baze podataka se može podesiti na PostgreSQL bazi putem konfiguracionih fajlova.

4. *Keširanje podataka*

Skladištenje podataka korišćenjem cache memorije pogodno je koristiti kada je potrebno čuvati veliku količinu podataka.

U našem sistemu, na ovaj način bi bilo pogodno čuvati podatke kao što su podaci o administratorima, lekarima, medicinskim sestrama, pacijentima, klinikama, salama, tipovima pregleda, slobodnim pregledima, lekovima, dijagnozama, itd. (podaci koji se rede menjaju).

Nasuprot njima, ne bi bilo pogodno na ovaj način čuvati podatke kao što su, na primer, podaci iz radnog kalendara, koji se često menjaju.



Slika 3 Keširanje podataka

5. Procena hardverskih resursa

Ukoliko je početna pretpostavka da naš sistem koristi 200.000.000 korisnika, procena hardverskih resursa za period od 5 godina izgleda ovako:

Entitet	Broj redova u narednih 5 godina	Zauzeće memorije (kB)	Zauzeće memorije (GB)
Admin	60 000	10 000	0.01
ClinicCenterAdmin	600	100	0.0001
Clinic	20 000	2500	0.0025
Doctor	240 000	60 000	0.06
Patient	199 579 400	40 000 000	40
Nurse	120 000	30 000	0.03
Medicine	500	500	0.0005
Diagnosis	1500	1500	0.0015
Recipe	20 000	30 000	0.03
Report	5 000 000 000	750 000 000	750
Examination	10 000 000 000	3 000 000 000	3000
ExaminationType	100	300	0.0003
EmergencyRoom	160 000	380 000	0.38
Schedule	10 054 000 000	800 000 000	800
Doctor_Patient	299 369 100	320 000 000	320
Grade	25 000 000	40 000 000	40
MedicalRecord	140 000 000	300 000 000	300

Tabela 1 Procena hardverskih resursa za date tabele koje predstavljaju okvirno 50% zauzeća baze podataka

6. Load balancing

Zahvaljujući komunikaciji između aplikativnih servera i Load balancer-a, server bi bio u mogućnosti da šalje informacije o svom zauzeću. Load balancer bi na osnovu tih informacija odlučivao kom serveru će sledećem poslati zahtev. Pored toga, ako bi se uzela u obzir i geografska lokacija čvorova (Traffic Routing), dobili bismo najbolje rešenje jer bismo na taj način vodili računa i o udaljenosti i o opterećenju samih servera.

7. Operacije koje treba nadgledati u cilju poboljšanja sistema

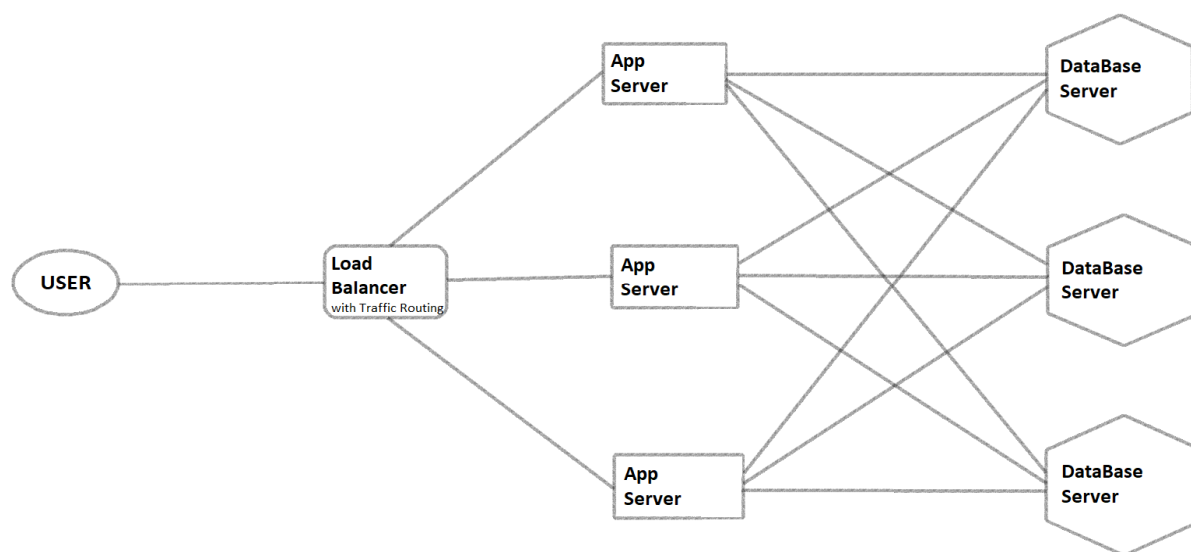
U cilju poboljšanja sistema, potrebno je pratiti i nadgledati način na koji ga korisnici koriste. Ukoliko posmatramo sve poslovne funkcionalnosti koje naš sistem obuhvata, dolazimo do zaključka da je od najveće važnosti posvetiti pažnju aktivnostima pacijenata, lekara i administratora.

Ukoliko posmatramo korišćenje sistema iz ugla pacijenta, treba što je više moguće skratiti i olakšati proces zakazivanja pregleda. Pored mogućnosti zakazivanja već definisanih pregleda, naš sistem pacijentu omogućava zakazivanje pregleda i drugim putem, pretragama i filtriranjem opisanim u specifikaciji projekta. Ovaj način zakazivanja je kompleksan i pogodan za unapređivanja, pogotovo kad je u pitanju korisnički interfejs.

Sa druge strane, kod korišćenja sistema iz ugla lekara, ono o čemu treba povesti računa je njegov način rada sa radnim kalendarom. Preglednost i ažurnost je izuzetno bitna, uzimajući u obzir da lekar u radnom kalendaru treba da ima uvid u svoje zakazane preglede, operacije, godišnje odmori i odsutne dane.

Na kraju, od izuzetne važnosti za ceo sistem su funkcionalnosti koje obavlja administrator, kako kliničkog centra, tako i klinike. Administrator se bavi održavanjem celokupnog sistema. Upravo iz tog razloga je bitno posvetiti pažnju unapređivanju i nadgledanju operacija koje obavlja. Ukoliko je to potrebno, treba skratiti put do određenih stranica i poboljšati izgled korisničkog interfejsa kako bi se ubrzao i olakšao posao koji obavlja.

8. Dizajn predložene arhitekture



Slika 4 Predložena arhitektura