

# Skalabilnost

## Pretpostavke:

ukupan broj korisnika aplikacije je 200 miliona,  
broj zakazanih novih pregleda i operacija na mesečnom nivou je milion,  
sistem moram biti skalabilan i visoko dostupan.

### 1. Strategija za particionisanje podataka - horizontalno

Razmatranjem mogućih opcija za particionisanje podataka došli smo do zaključka da nam je najkorisnije horizontalno particionisanje iz razloga što za određene tabele očekujemo brz rast perzistentnih torki. Razlog tome jeste što se na jednostavan i brz način može uklanjati zastareli podaci čija perzistencija više nije od značaja. S obzirom na veliki broj korišćenja SELECT i DML naredbi sa klauzulom WHERE ova vrsta perzistencije nam drastično ubrzava izvršavanje svih naredbi.

### 2. Predlog strategije za replikaciju baze i obezbeđivanje otpornosti na greške

Za obezbeđivanje otpornosti na greške predlažemo da pored primarnog servera baze postoji i backup server koji bi čuvao aktuelne podatke i služio za oporavak sistema u slučaju pada primarnog servera. Time bismo obezbedili minimalan gubitak podataka i pad sistema. Radi manjeg opterećivanja sistema predlažemo razdvajanje baze na dva servera, gde bi se jedan služio za skladištenje aktuelnih zahteva za registraciju pacijenata i traženje godišnjeg odmora. Drugi server bi se koristio za poslovanje klinike.

### 3. Predlog strategije za keširanje podataka

S obzirom na potencijalno veliku količinu podataka bilo bi veoma korisno koristiti keširanje, jer bismo time smanjili upite ka bazi i poboljšali performanse. Keširanje bismo koristili tako što bismo prvo proveravali da li traženi objekat postoji u keš memoriji, pa tek ako ne postoji slali bismo zahtev bazi. Time bismo u keš memoriju stavili one objekte za koje se najčešće šalju upiti. Konkretno, stavili bismo one podatke koje su pacijentu od velikog značaja, odnosno, za koje on najčešće šalje upite. To su njegov medicinski karton, podaci o klinikama i njihovim lekarima. Takođe bismo keširali i podatke o pregledima za koje su se najviše slali upiti u poslednjih mesec dana.

### 4. Okvirna procena za hardverske resurse potrebne za skladištenje svih podataka u narednih 5 godina

Pod pretpostavkom da svaki pacijent u bazu na mesečnom nivou unese 4.5MB podataka i da oni čine 95% korisnika od 200 miliona, za njih je na mesečnom nivou potrebno uskladištiti oko 855TB. Ostala količina memorije koja se zauzima tokom rada ostalih korisnika je zanemarljiva u odnosu na pacijenta. Za 5 godina potrebna količina bi bila 51.3PB( $855\text{TB} \cdot 12\text{meseci} \cdot 5\text{godina}$ ).

5. Predlog strategije za postavljanje load balansera

Korišćenjem load balansera između mreže i aplikacije kao i između keš memorije i servera smatramo da će se smanjiti opterećenje između navedenih slojeva arhitekture tako što će load balanser preraspoređivati pristigle zahteve između slojeva. Kao tehniku koju će koristiti load balanser predlažemo "Weighted Round Robin". Ova tehnika se zasniva na tome da će load balanser za svaki pristigli zahtev poslati serveru koji je sledeći po redu i koji ima najmanje zahteva. U cilju smanjenja saobraćaja biramo server koji ima manji broj zahteva u odnosu na ostale.

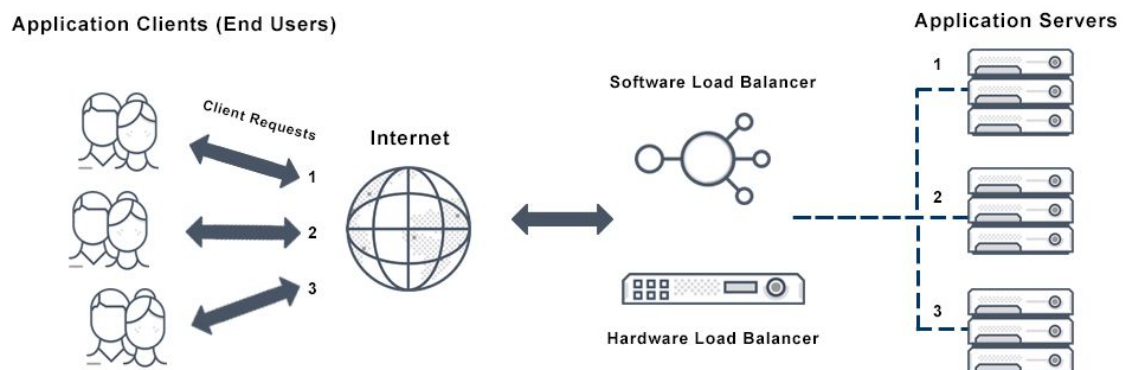
6. Predlog koje operacije korisnika treba nadgledati u cilju poboljšanja sistema

U cilju poboljšanja sistema potrebno je nadgledati skupe i česte operacije.

Navešćemo operacije koje bi trebalo nadgledati:

- i. Zahtevi za registraciju korisnika
- ii. Istoriju pregleda i operacija
- iii. Dodeljivanje sale za pregled
- iv. Zakazivanje pregleda ili operacije

7. Kompletan crtež dizajna predložene arhitekture (aplikativni serveri, serveri baza, serveri za keširanje, itd)



NAPOMENA: Uz pdf priložena je i slika Class djiagrama(ClassDiagram.png)