

Proof Of Concept **skalabilnosti aplikacije**

ISA/MRS 2022

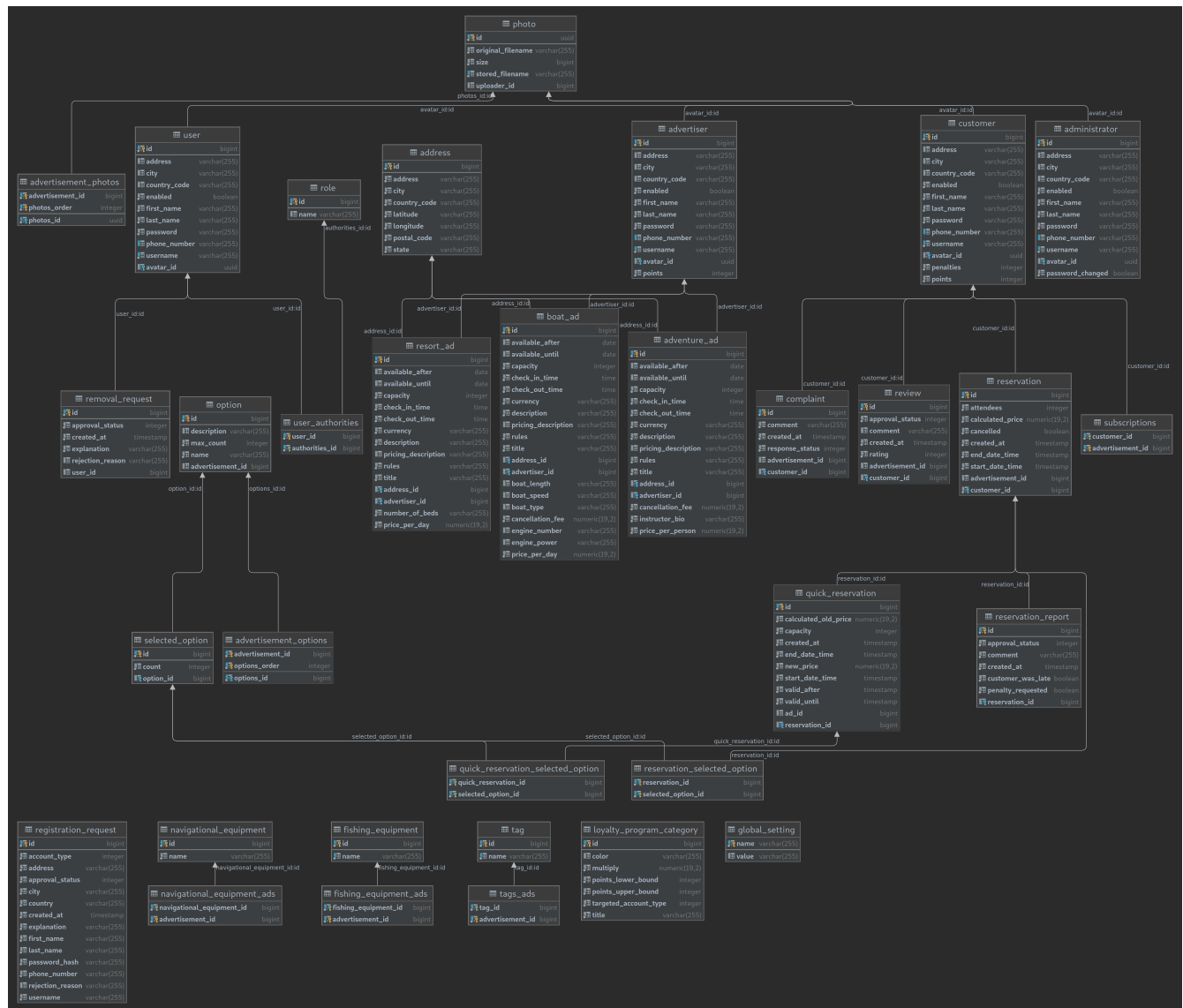
Tim 04

Dušan Lazić SW 4 2019

Ana Anđelić SW 14 2019

Milan Sekulić SW 54 2019

1. Dizajn šeme baze podataka



2. Predlog strategije za particionisanje podataka

Kako bismo poboljšali performanse naše aplikacije i osposobili je za obradu milion rezervacija mesečno, kao i za služenje ukupno sto miliona korisnika, potrebno je realizovati particionisanje podataka sa pažljivo odabranom strategijom. Dva često korišćena pristupa jesu horizontalno i vertikalno particionisanje. Smatramo da nam oba pristupa mogu biti od koristi.

Entiteti kojima se najčešće pristupa jesu oglasi (avanture, brodovi, odmarališta) i rezervacije, što ih čini polaznom tačkom kada su optimizacije u pitanju. Kako se rezervacijama najčešće pristupa radi utvrđivanja slobodnih termina, naš predlog je horizontalno particionisanje rezervacija po vremenu njihovog početka i kraja. Ukoliko bi, na primer, rezervacije bile raspoređene po mesecima, pretraga slobodnih termina bi bila znatno brža zbog manjeg prostora pretrage, ne bi bile razmatrane rezervacije u prošlosti, i sa porastom ukupnog broja rezervacija sistem ne bi postojao sporiji. Vertikalno particionisanje se takođe može primeniti, jer su za utvrđivanje slobodnih termina dovoljne samo dve kolone: početak i kraj rezervacije. Kako je ova operacija mnogo češća u odnosu na ostale operacije sa rezervacijama, vertikalnim particionisanjem možemo smanjiti suvišno čitanje ostalih kolona. Još jedna upotreba vertikalnog particionisanja može biti nad tabelom korisnika, kojoj se najčešće pristupa radi autentifikacije i autorizacije. Osim poboljšanja performansi, razdvajanje kredencijala i drugih osetljivih podataka u posebnu bazu pruža nam dodatne mogućnosti za poboljšanje sigurnosti sistema.

3. Predlog strategije za replikaciju baze i obezbeđivanje otpornosti na greške

U aplikacijama sa oglasima poput naše, svakoj izvršenoj rezervaciji prethodi razgledanje većeg broja oglasa. Štaviše, značajan broj korisnika ni ne napravi rezervaciju nakon razglednaja. Iz tog razloga možemo da očekujemo drastično veći broj operacija čitanja, nego operacija upisivanja u bazu. Kako je očekivani broj korisnika naše aplikacije sto miliona, potrebno je obezbediti brzinu i pouzdanost sistema.

Radi ubrzanja operacija čitanja i prevencije gubitka podataka u slučaju otkazivanja baze, predlažemo uvođenje redundantnih baza podataka. Imajući u vidu pomenuta očekivanja, predlažemo Master/slave arhitekturu, koja se sastoji od jedne *Master* baze kojoj se upućuju sve operacije pisanja, i više *Slave* baza kojima se upućuju sve operacije čitanja. Svaka operacija pisanja propagira se ostalim bazama kako bi podaci bili sinhronizovani. U slučaju otkazivanja *Master* baze, njenu ulogu može privremeno preuzeti jedna od *Slave* baza kako bi sistem ostao u funkciji.

4. Predlog strategije za keširanje podataka

Kao što je već pomenuto, u našoj aplikaciji možemo da očekujemo znatno veći broj upita čitanja nego pisanja. Imajući u vidu da se velik deo podataka vrlo retko menja (npr. opisne informacije o oglasima), ovaj broj upita možemo redukovati keširanjem podataka.

Naša aplikacija podržava L1 nivo keširanja koji je podržan od strane Hibernate-a bez dodatne konfiguracije. Ipak, mi predlažemo upotrebu L2 nivo keširanja, kojeg smo i implementirali u našoj aplikaciji na mikro primeru. Za ovaj nivo keširanja potreban je eksterni provajder, a u našem primeru korišćen je EhCache. Podaci koje ima smisla keširati jesu sami oglasi, jer informacije o njima nisu sklone čestim promenama. Ukoliko ipak dođe do promene, potrebno je upisati promenu u keš, za šta predlažemo *read write* strategiju.

Takođe, predlažemo keširanje resursa koji nisu podložni čestim promenama, a njihovo dobavljanje traje dugo. Na primer, ukoliko generišemo veoma detaljne izveštaje koristeći stare podatke. U nekim slučajevima, promene koje nisu značajne se takođe mogu zanemariti, tako da bi se mogle keširati prosečne ocene oglasa koji imaju velik broj ocena — ne očekujemo nagle i drastične promene prosečne ocene.

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

Entitet	Procena prosečnog zauzeća (kB)	Očekivani broj instanci	Očekivano ukupno zauzeće (GB)
Korisnik	0	10^8	15
Rezervacija	~ 0.1	$12 * 5 * 10^6 = 6 * 10^7$	6
Oglas	~ 1	$3\% * Korisnika * 1.2 = 3.6 * 10^6$	3.6
Fotografija	0	$Oglasa * 3.5 + Korisnika * 30\% = 4.2 * 10^7$	84 000
Recenzija	0	$Rezervacija * 60\% = 3.6 * 10^7$	5.4
			= 84030 ~ 84 TB

U tabeli su prikazane procene prosečnog zauzeća najčešćih entiteta. Izostavljeni su entiteti čiji broj ne raste brzo i ne doprinosi značajno zauzeću (tagovi, oprema...), a usko vezani entiteti i slični entiteti su uračunati (npr. opcije kod oglasa, oglašavači i klijenti zauzimaju slično prostora, itd). Procena se oslanja na pretpostavku da su 3% korisnika oglašavači i da u proseku imaju 1.2 oglasa, oglas u proseku ima 3.5 fotografije, 30% korisnika ima profilnu fotografiju i u 60% slučajeva klijent ostavi recenziju nakon rezervacije.

Zaključujemo da nam je potrebno blizu **84 TB**, i da skoro 100% zauzeća čine samo fotografije.

6. Predlog strategije za postavljanje load balansera

Aplikaciji sa sto miliona korisnika neće biti dovoljan jedan server za obrađivanje svih zahteva. Zbog toga, potrebno je da opterećenje adekvatno rasporedimo na više servera, što ćemo ostvariti pomoću *load balancer*-a. Sa load balanserom, klijenti više ne upućuju zahteve serverima direktno, već load balanseru koji potom preusmerava zahtev jednom od naših servera vodeći se odabranom strategijom.

Kako bismo koristili sve servere ravnopravno, predlažemo upotrebu *least connections* strategije koja svaki pristigli zahtev dodeljuje onom serveru koji je imao najmanje aktivnih konekcija u momentu pristizanja zahteva.

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

Praćenje aktivnosti korisnika nam može u velikoj meri poboljšati sistem u više aspekata. Informacije koje bismo sakupili nam mogu pomoći da steknemo uvid u upotrebljivost naše aplikacije, zadovoljstvo klijenata, da li su performanse i pouzdanost zadovoljavajući, da li klijenti pronalaze ono što ih interesuje, šta može povećati profit, i drugo.

Smatramo da je od velikog značaja da klijenti mogu da dođu do oglasa koji ih interesuju, tj. da sistem uspeva da dovede adekvatne oglase do korisnika. Iz tog razloga posebno treba nadgledati korisničke navike pri korišćenju pretrage, sa ciljem da se pretraga učini što efikasnijom i “pametnijom”, kako bi se povećala naša zarada, zarada oglašavača i zadovoljstvo klijenata.

8. Kompletan crtež dizajna predložene arhitekture

