

PROOF OF CONCEPT - ISA

PRERADA TRENUTNO KODA

Prvi korak za skaliranje aplikacije koji bi preuzeli je sređivanje koda i optimizacija. Greške koje se mogu lako ukloniti u kodu mogu kasnije koštati mnogo resursa. Uz to bi pokušali što više da koristimo upite baze a što manje proceduralno da rešavamo problem. Zatim bi pomoću alata za analizu koda dodatno sredili kod da bude lako proširiv i čitljiv kako bi omogućili što bezbolnije održavanje. Refaktorisanje kod je takođe neizbežno zbog trenutne neuredne strukture klasa (jedan kontroler/servis odgovara za više entiteta). Još jedna stvar koju bi morali da izmenimo je granulacija funkcija, pošto postoje slučajevi gde odstupamo od *Single Responsibility* principa. Naravno, da bi mogli da pustimo softver u produkciju potrebno je povećati broj testova radi garancije kvaliteta.

Mehanizam optimističkog zaključavanja bi poboljšao performanse prilikom akcija kao što su rezervacije mesta, automobila, sobe itd. koje moraju da opslužuju veći broj korisnika odjednom. Jedan od bitnijih mesta za poboljšavanje skalabilnosti su upiti. Omogućavanje korisniku da formira upit sa parametrima i stranicom, rezultovalo bi u mnogo boljim performansama pretraga.

Još jedna mogućnost bi bila da se radnici podele u pulove i da se tako raspoređuju resursi servera. Prilikom dodele poslova radnike bi organizovali u redove čekanja u slučaju preopterećenja servera. Nakon implementacije takvog mehanizma bi ostavili zahtevne operacija da se rade „offline“, kao što su proračuni prihoda u interval. Administratori bi videli podatke koju su stari nekoliko minuta (onoliko koliko je bilo potrebno da se sračuna) ali bi znatno smanjilo zagušenje servera.

SKALIRANJE I OPTIMIZACIJA SKLADIŠTENJA PODATAKA

Prvi korak bi nam bio vertikalno skaliranje na osnovu zahteva naše aplikacije i rezultata alata za nadgledanje resursa. Trenutna procena je da memorija i brzina pisanja diska bi bili prvi *bottle neck* na koji bi naišli zbog količine podataka koje korisniku treba da prikažemo i sačuvamo. Naravno, vertikalno skaliranje pomaže samo do određene mere i nakon njega treba implementirati i horizontalno skaliranje. Veoma je bitno uspostavljanje pravilne hosting topologije gde bi naš domen uvek pokazivao na load balanser koji bi omogućio da započnemo horizontalno skaliranje.

Zbog velike količine mejlova koje bi trebali da šaljemo sa 1.000.000 korisnik, neophodno je implementirati sopstveni smtp server. Način na koji trenutno predstavljamo lokacije pomoću *Yandex* mapa je još jedno ograničenje koje bi mogli da zaobiđemo pretplatom na *Google Maps*. Takođe, uvođenjem *Dynamic Content Caching*-a smanjili bi dodatno komunikaciju sa serverom. Pozitivna stvar koja nam olakšava skaliranje je činjenica da smo razvili aplikaciju po REST principu.

Relacione baze nisu najbolje rešenje za horizontalno skaliranje aplikacija, umesto njih bi prešli na NoSQL baze kao što su MongoDB. MongoDB baza podataka je veoma povoljna za horizontalno skaliranje uz pomoć *Sharding* mehanizma. *Sharding* je način distribucije podataka na više servera tako što se baza aplikacije podeli na *shard*-ove i korisniku se omogući pristup pomoću *mongosa*. *Mongos* je

query router koji predstavlja interfejs između klijentske aplikacije i baze podataka. Uz pomoć ovog pristupa horizontalno skaliranje ne bi predstavljalo nikakav problem. S' druge strane redovno održavanje baze podataka bi bilo neophodne zbog ovog mehanizma radi očuvavanja konzistentnosti podataka.

Što se tiče same baze, uveli bi indekse za najčešće tražene stvari tipa gradove, tipove automobila, servise u hotelu itd. Pored toga, keširanje čestih upita upita, kao što su letovi za Njujork, bi takođe poboljšalo performanse aplikacije.

OPTIMIZACIJA KLIJENTSKE APLIKACIJE

Prvu stvar koju bi poboljšali ovde je grupisanje upita i smanjenje njihovog broja. Primera radi za podatke administratora avio kompanije, na profilu aviokompanije možemo sve upite da grupišemo u jedan i time izbacimo nepotrebne pozive. Još jedna mogućnost je prelazak na framework koji bi poboljšao rad sa klijentskom aplikacijom. Uz pomoć *HTTP Caching*-a bi dodano mogli da smanjimo komunikaciju između klijenta i servera. Zbog velike količine statičkih fajlova koje sadrži naša aplikacija *Content Delivery Network* bi bilo solidno rešenje za distribuciju istih.

ZAKLJUČAK

Osim prethodno navednih rešenja, potrebno je obezbediti dovoljno dobar hosting i protok. U svakom slučaju, skaliranje naše veb aplikacije nije uopšte trivijalna stvar ali uz dovoljno truda i vremena je moguće osposobiti istu za podržavanje velikog broja korisnika.