

KEŠIRANJE



KEŠIRANJE

2

◆ ŠTA JE KEŠIRANJE?

- Čuvanje aplikativnih podataka na lokaciji kojoj je obezbeđen optimizovan pristup za brže korišćenje u odgovarajućem sloju n-slojne arhitekture.

◆ KEŠIRANJE NIJE UVEK NEOPHODNO

- Iako postoje čitave kategorije kandidata za keširanje, ti podaci se ne moraju nužno keširati jer sama izvedba može biti zahtevna, skupa ili može negativno uticati na druge operacije koje barataju tim podacima



SCENARIJI ZA KEŠIRANJE

◆ KORIŠĆENJE SKUPIH RESURSA

- Bilo koja operacija koja je skupa u pogledu korišćenih resursa (memorija/CPU/propusni opseg) može biti kandidat za keširanje
 - Podaci dobijeni čitanjem iz baze podataka
 - Fajlovi i binarni sadržaj dobijen sa udaljenih servera
 - Izveštaji dobijeni iz sistema za generisanje izveštaja
 - Pozivi internih i eksternih web servisa



SCENARIJI ZA KEŠIRANJE

◆ RESURSI KOJI PREDSTAVLJAJU USKO GRLO SISTEMA

- Neke softverske ili hardverske komponente mogu da postanu usko grlo sistema pod velikim opterećenjem
 - Podaci dobijeni od 3rd party proizvođača koji nudi servis sa jednog servera
 - Biblioteka ili stylesheet fajl koji se učitava uvek sa iste lokacije proizvođača

◆ POŠTOVANJE STRIKTNIH UGOVORA PO PITANJU PERFORMANSI (SLA)

- Neke operacije koje utiču na biznis aktivnosti i direktan prihod mogu biti podložni posebnim ugovorima
 - Početna ili neka druga specifična stranica bitna za biznis mora da se učitava za određeni broj sekundi
 - Transakcija plaćanja mora da se obavi za do 30-45 sekundi

◆ OPTIMIZACIJE PRIMENJIVE NA VIŠE TIPOVA UREĐAJA

- Optimizacije koje trebaju da uključe i operacije koje se mogu izvršavati na mobilnim uređajima a ne samo u web pretraživačima



SCENARIJI ZA KEŠIRANJE

6

◆ ČESTO POZIVANI SERVISI ILI KORIŠĆENI PODACI

- Sve operacije čiji pozivi rezultuju istim podacima dobri su kandidati za keširanje
 - Često korišćene vrednosti dobijene kompleksnim proračunima
 - Često korišćeni podaci iz baze podataka
 - Fiksne liste podržanih jezika, država, opcija na nivou aplikacije

◆ STATIČKI PODACI

- Bilo koji podaci koji nisu zavisni od korisnika i njegovog konteksta korišćenja dobri su kandidati za keširanje
 - Sadržaj koji ide u header i footer stranica
 - FAQ stranica
 - Kontakt stranica
 - ...



UTICAJ KEŠIRANJA

7

◆ UTICAJ NA SKALABILNOST APLIKACIJE

- Različitim tehnikama keširanja se mogu rešiti problemi koje unose komponente koje predstavljaju usko grlo sistema (prefetch, on-demand fetch)
- Ako postoje integracione tačke sistema koje nisu podložne skaliranju, keširanjem rezultata koji se dobijaju kroz njih može se minimizovati njihov uticaj na celokupnu arhitekturu
- Keširanje se može upotrebiti na nivou web servera, baze podataka, servisa, itd. i problem skalabilnosti se rešavati na nivou individualnih slojeva aplikacije



UTICAJ KEŠIRANJA

8

◆ UTICAJ NA PERFORMANSE APLIKACIJE

- Najveća prednost keširanja se ogleda u performansama aplikacija
 - Smanjuje se vreme za pravljenje poziva ka bazi podataka, web servisu, itd.
 - Keširanje rezultata kompleksnih operacija smanjuje ukupno utrošeno vreme za računanje
 - Keširanjem fajlova izbegava se ponovno dovlačenje istih preko mreže, kreiranje, parsiranje, itd.



UTICAJ KEŠIRANJA

◆ UTICAJ NA DOSTUPNOST APLIKACIJE

- U aplikacijama koje se zasnivaju na višeslojnoj arhitekturi, dostupnost se zasniva na dostupnosti svake komponente koja učestvuje u obavljanju biznis operacija
 - Ako je baza podataka ili mreža nedostupna, keširanje može pomoći da se privremeno nastavi neometan rad



ŠABLONI ZA KEŠIRANJE

10

- ◆ **KEŠIRANJE UNAPRED**
 - Podaci se keširaju kada se pokrene aplikacija
- ◆ **SCENARIJI KORIŠĆENJA**
 - Koristi se kada se trebaju keširati vrednosti koje su globalno dostupne u aplikaciji
 - Koristi se kada se vrednosti trebaju koristiti na početku komunikacije
- ◆ **KANDIDATI ZA KEŠIRANJE**
 - Konfiguracija aplikacije
 - Statičke vrednosti poput fajlova taksonomije sajta



ŠABLONI ZA KEŠIRANJE

11

- ◆ **KEŠIRANJE NA ZAHTEV**
 - Podaci se keširaju kada postoji eksplicitan zahtev za njima
 - Svi naredni zahtevi čitaju iz keša te podatke
- ◆ **SCENARIJI KORIŠĆENJA**
 - Koristi se kada nije garantovano da će se podaci koristiti u svim zahtevima
 - Koristi se kada je potrebno odloženo pribavljanje dodatnih vrednosti (lazy load)
- ◆ **KANDIDATI ZA KEŠIRANJE**
 - Lista država
 - Lista opcija
 - Paginirane vrednosti na stranici rezultata pretrage
 - ...



ŠABLONI ZA KEŠIRANJE

12

- ◆ **PREDIKTIVNO KEŠIRANJE UNAPRED**
 - Pravi se predikcija koji će se objekti keširati na osnovu ponašanja korisnika tokom korišćenja aplikacije
- ◆ **SCENARIJI KORIŠĆENJA**
 - Koristi se kada je izbor podataka koji će se keširati zasnovan na korisničkoj sesiji
- ◆ **KANDIDATI ZA KEŠIRANJE**
 - Statički elementi npr. slike, video zapisi, itd.



ŠABLONI ZA KEŠIRANJE

13

◆ KEŠIRANJE POMOĆU PROKSI SERVERA

- Koristi se zaseban server kako bi se ubrzalo slanje objekata

◆ SCENARIJI KORIŠĆENJA

- Može se koristiti uvek kad se razmišlja o skaliranju aplikacije

◆ KANDIDATI ZA KEŠIRANJE

- Statički fajlovi npr. slike, video zapisi, JS fajlovi, itd.

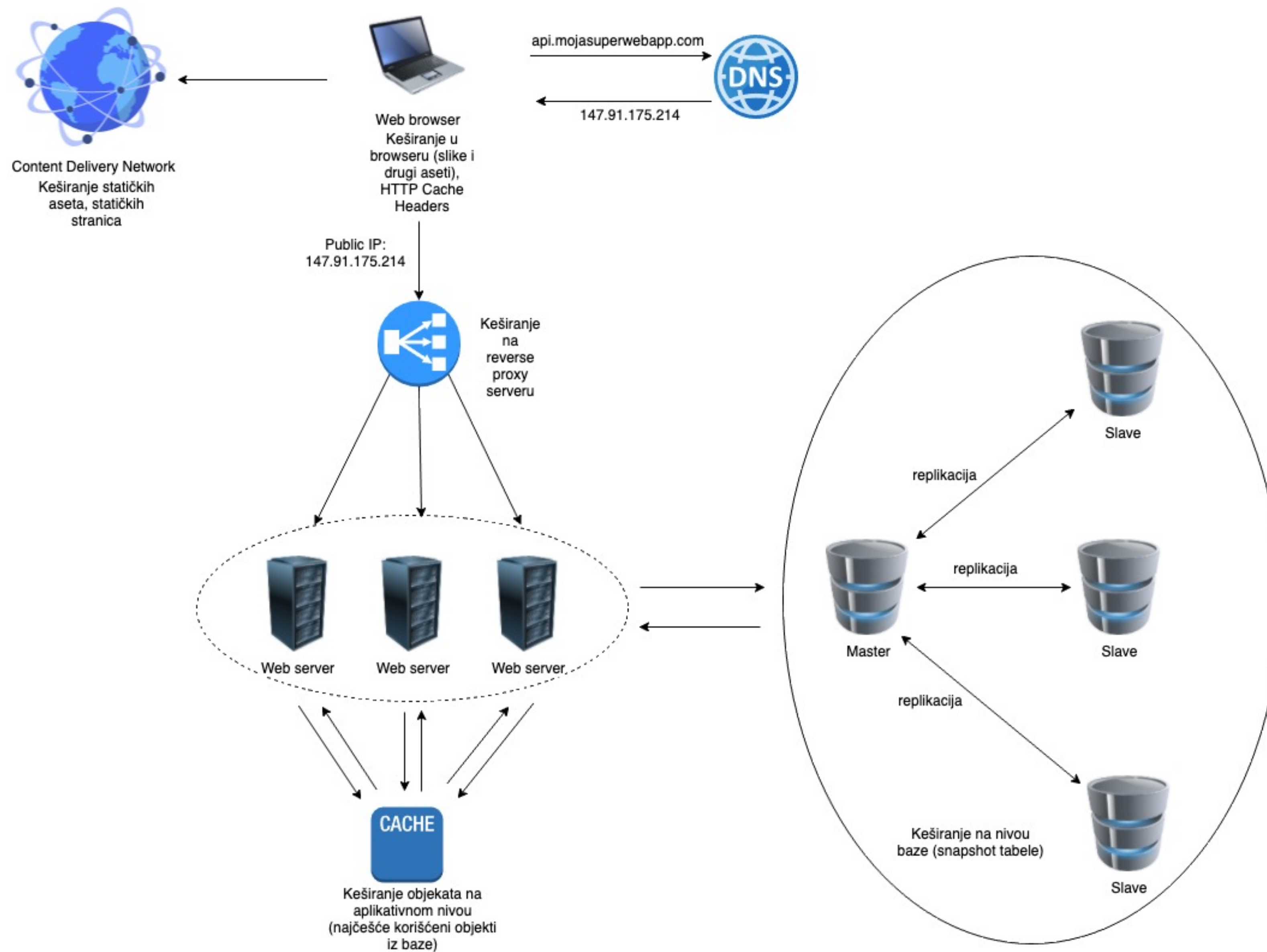


ŠABLONI ZA KEŠIRANJE

14

- ◆ **CONTENT DELIVERY NETWORK (CDN)**
 - Korišćenje geografski distribuiranih servera koji keširaju podatke
- ◆ **SCENARIJI KORIŠĆENJA**
 - Koristi se kada treba da se dodatno poboljšaju performanse optimizovanjem pristupa na osnovu geografske lokacije
- ◆ **KANDIDATI ZA KEŠIRANJE**
 - Statički fajlovi npr. slike, video zapisi, JS fajlovi, itd.

KEŠIRANJE NA VIŠE SLOJEVA





ANTIŠABLONI ZA KEŠIRANJE

16

◆ POGREŠNA SELEKCIJA KANDIDATA ZA KEŠIRANJE

- Primeri podataka koji se ne bi trebali keširati
 - "Dinamički podaci" – podaci koji se često menjaju te sistem mogu dovesti u nekonzistentno stanje (npr. količina proizvoda na stanju, cene, itd)
 - Lični podaci – podatke o korisniku kojima bi mogli pristupiti i drugi korisnici
 - Poverljivi biznis podaci – podaci koji bi trebali biti dostupni samo nalogizima koji imaju specifične uloge u sistemu
 - Veliki ili ugnježdeni objekti – podaci koji mogu da potroše mnogo memorije (npr. čitava kolekcija stranica aplikacije)



◆ DVE STRATEGIJE ZA INVALIDACIJU KEŠA

- Sistemi za keširanje bi trebali da obezbede načine za brisanje keša
 - Invalidacija bazirana na vremenu – za keš ili delove keša konfiguriše se TTL (time-to-live) i/ili TTI (time-to-idle)
 - Invalidacija na eksplicitan poziv

◆ STRATEGIJE ZA IZBACIVANJE KANDIDATA IZ KEŠA

- LFU (Least Frequently Used)
- LRU (Least Recently Used)
- FIFO (First In First Out)
- Custom?



METRIKE ZA PRAĆENJE

18

◆ ŠTA PRATITI?

- Cache hit ratio – što je veći broj to znači da je strategija bolja
 - $\text{Cache hit ratio} = \frac{\text{ukupan broj pogodaka}}{\text{ukupan broj zahteva}}$
- Cache miss ratio – što je veći broj znači da je strategija lošija
 - $\text{Cache miss ratio} = 1 - \text{Cache hit ratio}$
- Cache size – ukupna memorija koju koristi keš



REFERENCE

19

- ◆ SHIVAKUMAR K. S. ARCHITECTING HIGH PERFORMING, SCALABLE AND AVAILABLE ENTERPRISE WEB APPLICATIONS.
<https://doi.org/10.1016/B978-0-12-802258-0.00004-4>
- ◆ MOZILLA. HTTP CACHING. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching>
- ◆ SQUID CACHE <http://www.squid-cache.org/>
- ◆ AMAZON. CACHING OVERVIEW. <https://aws.amazon.com/caching/>
- ◆ SERVICE WORKERS <https://developers.google.com/web/fundamentals/primers/service-workers>
- ◆ EHCACHE <https://ehcache.org>

**KOJA SU VAŠA
PITANJA?**