

# NoSQL baze podataka

Predavanje 3: Key-value, Keširanje



**Univerzitet u Novom Sadu**  
**Fakultet Tehničkih Nauka**

# Keširanje podataka

- ▶ Keširanje podataka je proces koji skladišti više kopija podataka ili datoteka na privremenoj lokaciji za skladištenje — **cache**
- ▶ Cache čuva podatke za softverske aplikacije, servere i web pretraživače, ali i za sistemski software
- ▶ Ovaj proces osigurava da korisnici ne moraju da preuzimaju informacije svaki put kada pristupe web lokaciji ili aplikaciji
- ▶ Čitava ideja je nastala oko ideje: kako bi ubrzali učitavanje sajta, ili što pre pristupili podacima

- ▶ Čitanje podataka sa diska je (relativno) sporo, pogotov kada imate dosta podataka
- ▶ Razmišljajte na nivou Facebook, Instagram, Amazon, ...
- ▶ Sa druge strane disk je relativno jeftin, i ima ga prilično dosta
- ▶ Čitanje podataka iz memorije je (relativno) brzo
- ▶ Ali opet, memorije nemamo toliko puno — ma šta neko rekao
- ▶ U memoriji su obično najsvežiji podaci

- ▶ Cena nije ista kao cena diska, a ni kapacitet
- ▶ Plus ako se sistem restartuje nema više podataka
- ▶ U memoriji čuvati samo podake kojima se (relativno) skoro pristupalo
- ▶ Ne čavi sve podatke!
- ▶ Ovo će omogu'čiti brži pristup podacima
- ▶ Ali moramo naći način da čuvamo najnovije informacije

- ▶ Postoje razni načini i strategije kako čuvati podatke u cache-u
- ▶ Možemo da rotiramo sadržaj, ako je struktura fiksne dužine
- ▶ Imamo nekakav fiksni skup, koliko elemenata može da uvek bude u cache-u
- ▶ Noviji elementi izbacuju one starije
- ▶ Možemo definisati život podataka u cache-u (time to live — TTL)
- ▶ ...

# Pitanje 1

Koliko dugo čuvati podatke u kešu, kada ih izbacivati...

ideje :)?

- ▶ Tačan odgovor ne postoji, ili bar meni nije poznat :)
- ▶ Jedna strategija se ne može primeniti na sve slučajeve
- ▶ Zavisi od aplikacije
- ▶ Zavisi od slučaja korišćenja
- ▶ Zavisi od resursa
- ▶ ...

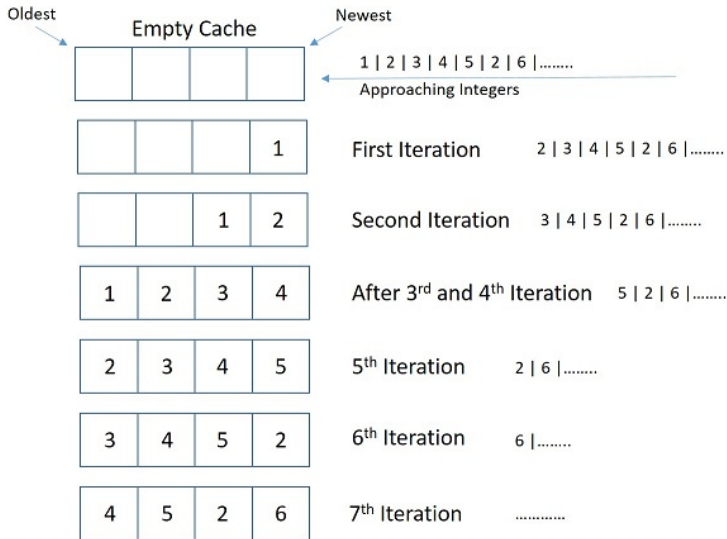
## Cache — Zahtevi

- ▶ Cache sistemi obično imaju zahteve koje treba da isputene, i neki od njih su:
  - ▶ Fiksna veličina: Cache memorija treba da ima neke granice, tj. da ograniči upotrebu memorije
  - ▶ Brz pristup: Operacija dodavanja elementa u cache i traženja treba da bude brza, poželjno  $\mathcal{O}(1)$  vreme
  - ▶ Zamena unosa: U slučaju da je dostignuto ograničenje memorije, cache treba da ima efikasan algoritam za izbacivanje unosa kada je memorija puna
- ▶ Dodavanje ograničenja je obično eksternizovano u nekakvu konfiguraciju — konfiguracioni fajl, ili neki drugi oblik



# LRU

- ▶ Najmanje nedavno korišćen (LRU) organizuje elemente po redosledu korišćenja
- ▶ Omogućavajući da brzo identifikujemo koja stavka nije korišćena najduže vreme
- ▶ Zamislite stalak za odeću, gde je odeća uvek okačena na jednoj strani
- ▶ Da bi pronašli predmet koji smo najmanje koristili, pogledate predmet na drugom kraju staka
- ▶ Ukratko, to je politika izbacivanja iz keša
- ▶ Navodimo kada se naš keš popuni i doda se novi element, uklanjamo najmanje nedavno korišćenu stavku iz keša



(Opgenus, Implement Least Recently Used (LRU) Cache)

► Prednosti:

- Super brzi pristupi: LRU keš čuva stavke po redosledu od nedavno korišćenih do najmanje korišćenih — oba mogu pristupiti u  $\mathcal{O}(1)$  vremenu
- Super brza ažuriranja: Svaki put kada se pristupi stavci, ažuriranje keša traje  $\mathcal{O}(1)$  vremena

► Mane:

- LRU keš koji čuva  $n$  stavki zahteva spregnutu listu dužine  $n$ , i hash mapu koja sadrži  $n$  stavki — to je  $\mathcal{O}(n)$  prostor, ali to su i dalje dve strukture podataka (za razliku od jedne)

# TTL

- ▶ Key-value model ima i *Time to Live (TTL)* mehanizam
- ▶ Odnosno vremenski čuva podatak
- ▶ Ovo može biti zgodno za implementaciju raznih funkcionalnosti vaših aplikacija
- ▶ Ova operacija nije tako jednostavna za implementirati u relacionoj bazi, i obično zahteva upotrebu još minimalno jednog alata
- ▶ Cela ideja iza ovog tipa podataka je da on bude aktivan neki fiksni vremenski interval  $T_d$ , nakon čega biva obrisan

- ▶ Neki sistemi za skladištenje omogućavaju i mehanizam notifikacija kada TTL istekne
- ▶ Što može biti zgodno, za automatsko osvežavanje keša
- ▶ Kod key-value baza koje nisu striknot u memoriji, podržavaju i zapis na disk pored memorije, ovaj tip će biti perzistiran i na disk
- ▶ Ova osobina je vrlo korisna u slučaju da instanca postane nedostupna usled pada

- ▶ TTL je dosta korišćen mehanizam za keširanje podataka
- ▶ Internet pretraživači ga dosta koriste
- ▶ CDNs se oslanjaju na ovaj mehanizam za keširanje statičkog sadržaja
- ▶ Ali se dosta koristi i u aplikativnom pa i sistemskom softveru kao zgodan i vrlo jednostavan mehanizam keširanja podataka

## Dodatni materijali

- ▶ A Low Overhead High Performance Buffer Management Replacement Algorithm
- ▶ Policy with Applications to Video Streaming
- ▶ <https://hazelcast.com/glossary/cache-miss/>
- ▶ Data Caching in Cassandra

# Pitanja

Pitanja :) ?