



VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
DUOMENŲ MOKSLAS

## **Iliustracijų saugojimo servisas**

Situacijos analizė

Atliko: Mantas Bilaišis, Simona Gelžinytė

Vilnius  
2022

## Įvadas

LSP numeriai:  $(2013180 + 2013191) \bmod 6 = 5$

Gautai situacijai įvertinkite:

- Kokį duomenų modelį pasirinksite (reliacinį, rakto-reikšmės, dokumentų, stulpelių, grafų)?
- Kokiomis savybėmis pasižymėti turi duomenų bazė (saugojimo duomenų modeliui (Hash Table, B-Tree, LSM), plečiamumu, ACID garantijomis, ...)?
- Kokią iš žinomų duomenų bazių būtų tinkamiausia? Kokie jos trūkumai? Kokias savybes turėtų turėti ideali situacijai duomenų bazė.
- Pateikite fizinį duomenų modelį, pasirinktai duomenų basei (stulpelius, indeksus, raktus, duomenų tipus ir t.t.). Taip pat pateikite duomenų schemas sukūrimo sakinius ir duomenų įterpimo pavyzdžius.

Atlikdami užduotį įvertinkite alternatyvas. Pagrįstai atmesdami netinkamas, pagrįstai palikdami tinkamas. Kartais idealių sprendimų nėra, nepamirškite įvardinti pasirinkto sprendimo trūkumus.

Situacija 5 - Iliustracijų saugojimo servisas (pavyzdžiui [imgur.com](http://imgur.com)).

- Iliustracijų servisas leidžia visiems interneto vartotojams laisvai įkelti iliustracijas.
- Įkėlus iliustraciją servisas pateikia vartotojui nuorodą, kurią bus galima naudoti paveiksliukui parodyti.
- Paveiksliukai turi ribotą gyvavimo laiką – jie saugomi tik vieną dieną.
- Servisui nėra didelė bėda paveiksliukus prarasti ir po trumpesnio laiko.
- Serviso atsakymo laikas itin svarbus.

## Situacijos analizavimas

Pasirinktas duomenų modelis – rakto reikšmės (pvz. raktas - paveikslėlio id, reikšmė – sąrašas ar „hash“ struktūra su paveikslėliu, vartotojo id (jei sistema leistų vartotojams registruotis), paveikslėlio pavadinimu ir pan.). Raktas gali būti pateikiamas kaip unikalus identifikatorius arba unikalus kelias į reikšmę. Duomenys įrašomi (įterpiami, atnaujinami ir ištrinami) ir pateikiami užklausos pagal raktą, skirtą jų vertei išsaugoti/gauti. Rakto reikšmės duomenų bazė, susieja reikšmę (kuri gali būti bet kas, nuo skaičiaus ar paprastos eilutės iki sudėtingo objekto) su raktu, kuris naudojamas stebėti objektą. Rakto reikšmės duomenų bazėse naudojamos kompaktiškos, veiksmingos indeksų struktūros, kad būtų galima greitai ir patikimai rasti vertę pagal raktą, todėl jos idealiai tinka sistemoms, kurios turi sugebėti rasti ir gauti duomenis pastoviu laiku. Kadangi mums yra itin svarbus atsakymo laikas, rakto reikšmės duomenų bazė yra puikus variantas.

Dėl didelės greičio svarbos ir pasirinkto rakto-reikšmės duomenų modelio tinkamiausias duomenų saugojimo modelis yra Hash Table. Kadangi mums yra svarbus atsakymo laikas ir duomenų bazė gali prarasti paveikslėlius greičiau nei per nustatytą gyvavimo laiką, vieną dieną, duomenų bazė turi būti AP, kuri ir sutrikus ryšiui greitai atsakys į užklausą nebūtinai naujausiais duomenimis. Norimų atlikti operacijų vientisumas nėra būtinas, o sistemos prieinamumas svarbus, todėl duomenų bazė turi tenkinti BASE garantijas.

Tinkamiausia duomenų bazė – **Redis**.

Privalumai: greitas rašymas ir nuskaitymas, galima nustatyti gyvavimo laiką, po kurio duomenys bus ištrinti, galima saugoti didelės raktas-reikšmės poras (iki 512 MB), taip pat Redis gali saugoti ir dvejetainį raktas-reikšmės poras, taigi png/exe/pdf ir t.t. failai gali būti įrašomi į Redį kaip raktas-reikšmės.

Trūkumai: Redis objektus saugo atmintyje, o paveikslėliai gali užimti daug vietos, todėl jei kasdien bus keliama daug paveikslėlių, reikės daug atminties. Ši problema yra sprendžiama paveikslėlius suspaudžiant (compress), taip sumažinant jų dydį. Tarkime, kad į servisą per dieną įkeliama apie 1,5 milijono paveikslėlių (imgur.com 2015 m. duomenys <https://expandedramblings.com/index.php/imgur-statistics/>). Tarkime, kad suspausto paveikslėlio dydis yra 50 kB. Tokiu atveju reikės daugiau nei 70 GB atminties nekuriant jokių kopijų. Naudoti tokius didelius atminties kiekius yra brangu. Galimas šios problemos sprendimo būdas – pačius paveikslėlius saugoti ne duomenų bazėje, o saugykloje, tokioje kaip, Amazon S3, o duomenų bazėje saugoti nuorodas į paveikslėlius ir jų metaduomenis. Būtent taip realybėje veikia nemaža dalis iliustracijas saugančių platformų.

Kaip turėtų veikti tinkama duomenų bazė – įkeliama iliustracija, jai suteikiamas unikalus pavadinimas ir yra surenkama su įkeltu vaizdu susijusi informacija (pvz.: naudotojo vardas, kokiam albumui priklauso (jeigu priklauso)), ji išsaugoma pasirinktoje duomenų bazėje, mūsų atveju, Redis. Iliustracija yra išsaugoma debesio pavidalo saugykloje (pvz.: Amazon S3). Įkeltas vaizdas patenka į SAN (Storage Area Network, tai iš esmės būtų milžiniška standžiųjų diskų grupė, kurią galima suskirstyti į saugyklos dalis ir priskirti konkretiems serveriams). SAN yra sukurtas taip, kad kiekviena jo dalis turėtų kitą bitą, kuris gali perimti iliustraciją, jeigu pirmasis sugenda. Tad iliustracija išlieka, jei miršta diskai, jei miršta aparatinė įranga arba nutrūksta kabeliai.

Kodėl kitos žinomos duomenų bazės netinkamos?

- MongoDB gali saugoti paveikslėlius naudojant GridFS specifikaciją, paveikslėlių metaduomenis atrodo patogiau saugoti JSON formatu. Tačiau nors ir Mongo galima nustatyti objektų gyvavimo trukmę, ji automatiškai neišvalo vietos diske, todėl tai turi būti padaryta rankiniu būdu arba paleidus iš naujo programą. Kadangi mūsų atveju duomenis reikia valyti dažnai, šis variantas nėra optimalus.
- Cassandra netinka, nes joje sunku ištrinti duomenis. Įprastu atveju, Cassandra uždeda žymą duomenims apie jų pašalinimą, tačiau jie nėra ištrinami, tik negalima prie jų prieiti. Ji gali apdoroti labai daug duomenų ir daug užklausų, bet atsakymo greitis sulėtėja, o mums itin yra svarbus atsakymo laikas. Duomenys modeliuojami pagal užklausas, o ne pagal struktūrą, todėl ta pati informacija saugoma kelis kartus.
- Neo4J netinkama, nes mums nėra svarbūs ryšiai tarp kelių esybių. Neo4J taip pat yra sunkiai „scalable“, todėl sulėtėja duomenų įrašymai.
- Reliacinė duomenų bazė yra netinkama šiai problemai, nes mums labai svarbus serviso atsakymo laikas, o reliacinės duomenų bazės yra lėtesnės už nereliacines.

## Fizinis duomenų modelis

Kadangi šiame servise saugosime tik paveikslėlius ir nesaugosime jokios informacijos apie juos keliančius interneto vartotojus, duomenų bazėje užteks vienos esybės. Jos modelis pavaizduotas 1 pav. Raktas – paveikslėlio ID, unikalus kiekvienam paveikslėliui, UUID tipo. Kiti pasirinkti saugoti atributai – paveikslėlis, užkoduotas dvejetainė seka, nuoroda į paveikslėlį bei paveikslėlio pavadinimas (abu tekstinių duomenų tipo), būtų saugomi „hash“ struktūroje.

Images	
<u>Image ID</u> (Key)	UUID
Image	Bytes
Link	String
Title	String

1 pav. Fizinis duomenų bazės modelis

Duomenų įterpimo pavyzdys:

HSET ece48f51-a2e9-4150-9bb3-8f5492ebf206 Image pic.png Link "imageservice.com/ece48f51a2e941509bb38f5492ebf206" Title "Picture1"

Pavyzdys naudojant Python:

```
import redis
from datetime import datetime
import uuid
import time
import random

r = redis.Redis(host='localhost', port=6379)
r.flushdb()

user_id = random.getrandbits(32)
print("upload your illustration link: ")
png_id = str(input())
link = uuid.uuid4()
link = str(link.hex)

current_dateTime = datetime.now()

def data(r,user_id, png_id, link):
    r.hset(f'{user_id}:{png_id}', 'linkk', link)
    return None

def put_data(r, user_id, png_id, link):
    key = f'{user_id}:{png_id}'
    r.expire(key, 86400)
    print(f"Your illustration with the following link: {link} will expire in 24 hours")
```

```
print("left seconds till expiration: ", r.ttl(key))
time.sleep(10)
print("your link will expire in : ", r.ttl(key), " sec.")
return None
```

```
data(r, user_id, png_id, link)
put_data(r,user_id, png_id, link)
```