

Situacijos analizė

Situacija – iliustracijų saugojimo servisas

- 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.

Reikalavimai duomenų bazei

- Bendru atveju duomenų bazė turėtų pasižymėti greitu atsakymu į užklausas ir geru pasiekiamumu. Papildomai, esant dideliame duomenų kiekiui, svarbi ir plečiamumo savybė.
- Kadangi vartotojai paveiksliukus gali kelti laisvai, nebūtina laikyti ir tvarkyti vartotojų duomenų. Paprasčiausiu atveju duomenų bazės įrašai gali būti tiesiog failo ir nuorodos, į įkeltą failą, poros. Tuomet gaunamas paprastas duomenų bazės modelis, kuriam nereikia sudėtingų užklausų, duomenų agregavimo ar keitimo.
- Saugomi paveiksliukai turi ribotą gyvavimo laiką, todėl duomenų bazė turėtų palaikyti *time-to-live* funkcionalumą. Kadangi paveiksliukus nėra didelė bėda prarasti po trumpesnio laiko, esant dideliame duomenų kiekiui, seniausi paveiksliukai gali būti trinami.
- Serviso atsakymo laikas itin svarbus, todėl sistemai pasidalijus siekiama pasiekiamumo – asinchroninis replikavimas, o bendru atveju svarbiausias atsakymo greitis.
- Norimos duomenų bazės charakteristikos – PA/EL (pagal PACELC).

Duomenų bazių privalumai ir trūkumai

Šiam atvejui, iš nagrinėtų duomenų bazių (Redis, MongoDB, Cassandra, Neo4j) galima atmesti Neo4j, nes turimi duomenys nėra niekaip susieti sąryšiais, todėl grafų duomenų bazės modelio taikymas nėra prasmingas.

Redis

Privalumai:

- Rakto-reikšmės duomenų modelis, kur raktas yra nuoroda, o reikšmė – paveiksliuko failas (base64 ar binariniu formatu).
- Duomenys saugomi hash table struktūroje – greičiausias būdas skaityti duomenis turint raktą (kurį vartotojas gauna įkėlus paveiksliuką).
- Duomenų bazei nėra sudėtingų paieškos užklausų, reliacinių duomenų, duomenų agregavimo ar keitimo.
- Duomenų bazė saugo duomenis operatyviojoje atmintyje, todėl skaitymas/rašymas yra itin greitas.
- Palaiko TTL funkcionalumą – duomenys laikomi atmintyje vieną dieną, esant vietos trūkumui gali būti trinami seniausi įrašai.

- Greitas atsakymo laikas – *Active-Passive* modelis, asinchroninis replikavimas – rašoma į šeimininko mazgą, o skaitoma iš bet kurio.
- Galimas aukštas pasiekiamumas ir horizontalus plečiamumas su Redis Cluster topologija, kai sistemoje yra keli šeimininko mazgai galintys priimti rašymo bei skaitymo užklausas.

Trūkumai:

- Labai mažas atminties kiekis – kadangi duomenų bazė duomenis laiko operatyviojoje atmintyje, prieinamas atminties kiekis yra itin ribotas. Neribojant failų dydžių galima labai greitai pritrūkti atminties.
- Šiuo atveju padeda tai, kad laikomi failai galioja tik vieną dieną – turint papildomų resursų galima pridėti papildomų serverių/mazgų, kurių turėtų užtekti laikant duomenis ribotą laiką.
- Operatyvios atminties kaina yra daug didesnė lyginant su diske esančia atmintimi, todėl tokios duomenų bazės plėtimas kainuotų labai brangiai.

MongoDB

Privalumai:

- Galimas geras pasiekiamumas – *Active-Passive* modelis, kai nustojus veikti pirminiam mazgui, jo darbą gali perimti kitas. Asinchroninė replikacija, rašoma į šeimininko mazgą ir skaitoma iš šeimininko mazgo, bet galima sukonfigūruoti skaitymą iš bet kurio – geresnis atsakymo laikas.
- Horizontalaus plečiamumo galimybės.
- Duomenys saugomi diske, todėl nėra atminties trūkumo problemos, galima saugoti didesnius failus.
- Jei norima platesnio serviso funkcionalumo, dokumentinis duomenų bazės modelis leidžia patogiau laikyti su paveikslu susijusius duomenis (tags, description, user info, etc.).
- Jei norima platesnio serviso funkcionalumo, B-tree duomenų saugojimo struktūra leidžia efektyvią įrašų paiešką (pvz. agregavimas pagal kokią nors kategoriją, rikiavimas pagal peržiūrų skaičių).

Trūkumai:

- Dokumentinis duomenų bazės modelis nėra itin naudingas, nes yra tik du pagrindiniai elementai – nuoroda ir paveiksluko failas.
- TTL funkcionalumas palaikomas, bet reikalauja failo įkėlimo datos lauko ir susieto indekso.
- Duomenys saugomi B-tree struktūroje, kuri bendru atveju efektyvi, bet nusileidžia maišos lentelių implementacijai turint tiesioginę priėjimą prie duomenų (pirminį raktą – nuorodą).

Cassandra

Privalumai:

- Aukštas pasiekiamumas – *Peer-to-Peer* modelis, kur bet kuris mazgas gali priimti rašymo bei skaitymo užklausas, asinchroninė replikacija – prioretizuojamas atsakymo laikas.

- Horizontalaus plečiamumo galimybės, neįtakojant duomenų bazės sistemos veikimo.
- Lengviau implementuoti geografiškai išskirstytą sistemą, kuri dar labiau sumažina atsakymo laiką.
- Palaikomas TTL funkcionalumas.
- Stulpelių duomenų modelis su dviem laukais veiktų panašiai kaip rakto-reikšmės modelis.

Trūkumai:

- Paveikslėlius reikėtų saugoti kaip *blob* elementą duomenų bazėje, kurio dydžio limitas efektyviam užklausų vykdymui yra < 1MB, todėl net turint daugiau atminties failo dydžius reikėtų riboti.

*Paveikslėlių galima dalinti į kelis *blob* elementus (nesu tikras ar toks duomenų skirstymas paveikia užklausų atsakymo greitį).

- Dažniausiai naudojama dideliems duomenų kiekiams išskirstytose sistemose laikyti – šiuo atveju tikėtina, kad duomenų kiekis nebus didelis, nes paveikslėliai gyvuoja tik vieną dieną.
- Duomenys saugomi B-tree struktūroje, kuri bendru atveju efektyvi, bet nusileidžia maišos lentelių implementacijai turint tiesioginę priėjimą prie duomenų (pirminį raktą – nuorodą).

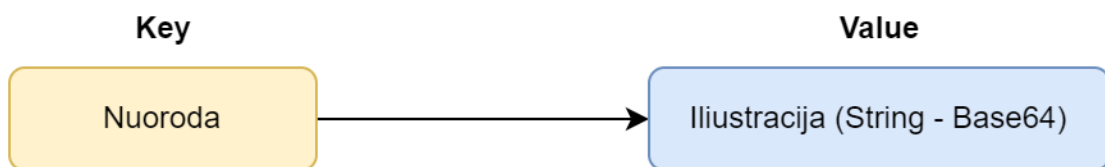
Tinkamiausia duomenų bazė

Paprastiausiam funkcionalumui, kai servisas atlieka tik paveikslėlių saugojimą, tinkamiausia duomenų bazė yra Redis. Redis puikiai tinka rakto-reikšmės duomenų struktūrai, turi itin greitą atsakymą į užklausas ir pasiekiamumo garantijas, su atitinkama sistemos konfigūracija.

Didžiausias trūkumas yra mažas prieinamos atminties kiekis, todėl būtų prasminga keliamų iliustracijų dydį riboti (priklausomai nuo turimos atminties).

Galiausiai šis sprendimas įmanomas tik todėl, kad paveikslėliai laikomi ribotą laiką – vieną dieną (su prielaida, kad šis servisas nėra end-all be-all failų saugojimui ir nesulaukia šimtų tūkstančių paveikslėlių per dieną).

Duomenų bazės modelis labai paprastas – rakto-reikšmės poros.



o duomenys įterpiami sakiniu – *SETEX key ttl_in_seconds value*.

Pavyzdžiui – *SETEX puKyhvE 86400 "/9j/2wBDAAgGBgcGBQgHBwcJCQgKD..."*