Razvoj SUPB z integracijo v programski jezik Python

Janez Sedeljšak
Mentor: doc. dr. Boštjan Slivnik
Somentor: asist. dr. Marko Poženel

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

js0578@student.uni-lj.si

September 6, 2023

Kazalo

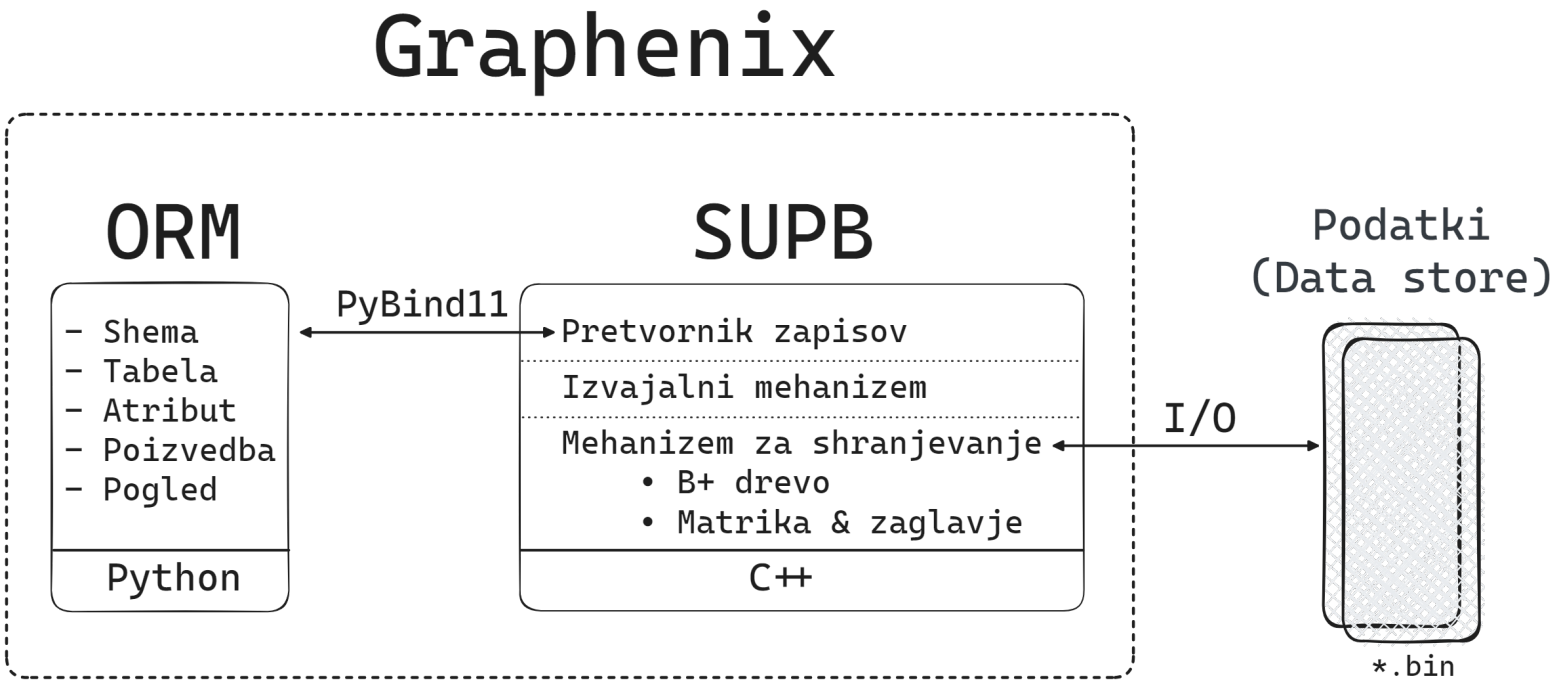
Motivacija

- Spoznati delovanje podatkovnih baz
- Izogib uporabi anti-vzorcev

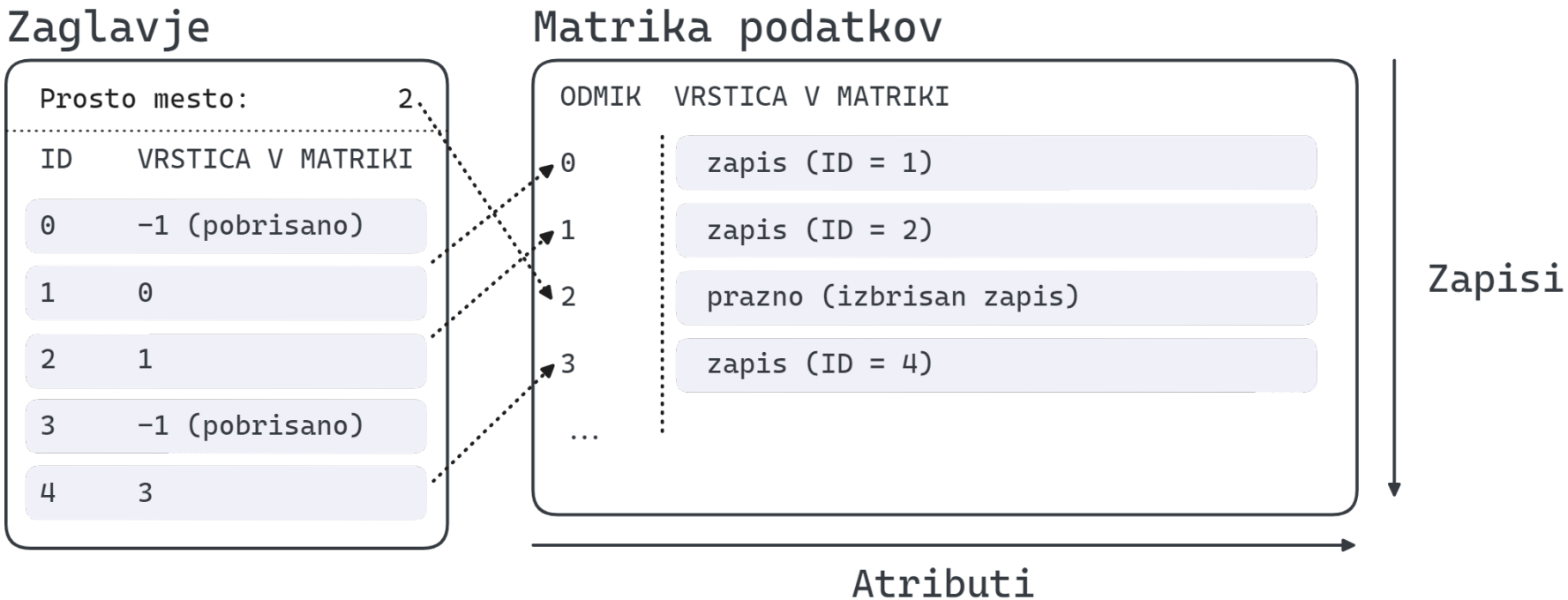
Glavni cilji:

- Razvoj minimalističnega SUPB za programski jezik Python:
  - SUPB na nivoju programskega jezika C++
  - Indeksiranje z uporabo B+ dreves
  - Intuitiven način komunikacije s podatkovno bazo
  - Delno primerljiv z SQLite

Arhitektura rešitve Graphenix

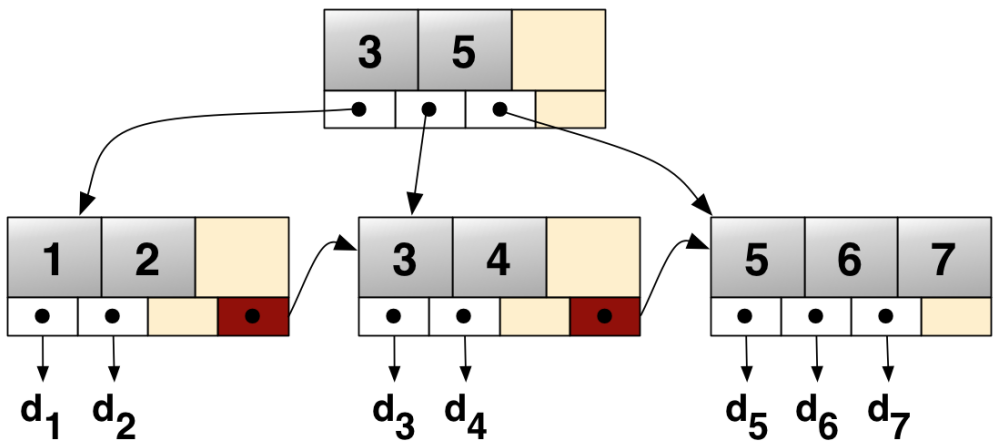


Struktura shranjenih podakov



Indeksiranje z uporabo B+ drevesa

- Implementacija s pomočjo programskega jezika C++
- Shranjevanje strukture v binarni datoteki
- Uproaba "generikov" za različne podatkovne tipe (nizi, cela števila, realna števila, povezave)



1. Gručanje zapisov ob branju matrike podatkov
2. Uporaba prioritetne vrste ob branju z omejevanjem in urejanjem
3. Uporaba vgnezdenih podatkovnih okvirjev za predstavitev rezultatov

Podatkovni tipi

```
enum FIELD_TYPE
{
    INT = 0,
    STRING = 1,
    BOOL = 2,
    DATETIME = 3,
    LINK = 4,
    DOUBLE = 5,
    VIRTUAL_LINK = 6
};
```

Operacije filtriranja

```
enum FILTER_OPERATION_TYPE
{
    EQUAL = 0,
    NOTEQUAL = 1,
    GREATER = 2,
    GREATER_OR_EQUAL = 3,
    LESS = 4,
    LESS_OR_EQUAL = 5,
    REGEX = 6,
    IS_IN = 7,
    NOT_IN = 8,
    BETWEEN = 9,
    IREGEX = 10
};
```

Definiranje sheme

```
class User(gx.Model):
    name = gx.Field.String(size=100)
    tasks = gx.Field.VirtualLink("user")
    sent = gx.Field.VirtualLink("sender")
    recieved = gx.Field.VirtualLink("reciever")

class Task(gx.Model):
    content = gx.Field.String(size=100)
    user = gx.Field.Link()

class Message(gx.Model):
    content = gx.Field.String(size=50)
    date = gx.Field.DateTime()
    sender = gx.Field.Link().as_index()
    reciever = gx.Field.Link().as_index()

my_schema = Schema('my_schema', models=[User, Task, Message])
my_schema.create(delete_old=True)
```

Poizvedovanje

```
# uporabniki s padajočo ureditvijo
_, view = User.order(User.name.desc()).all()

# imena uporabnikov z odmikom 5 in omejitvijo 10
users = User.offset(5).limit(10).pick(User.name)

# sporočila, ki jih je danes prejel uporabnik John
_, view = Message.filter(Message.reciever.equals(john)).all()

# število sporočil in datum zadnjega sporočila po uporabnikih
counts = Message.agg(by=Message.sender,
    count=gx.AGG.count(), latest=gx.AGG.max(Message.date))

# uporabniki in zadnja 3 prejeta sporočila
_, view = User.link(
    recieved=Message.order(Message.date.desc()).limit(3)
).all()
```

Sestavljanje pogojev

Nabor sporočil s sestavljanjem pogojev

Preberemo vsa sporočila, ki so bila poslana v zadnjih petih dneh.
Poleg tega zahtevamo, da je izpolnjen eden izmed pogojev:
Pošiljatelj/prejemnik je uporabnik `john` ali prejemnik ni `jane`

```
_, view = Message.filter(
    Message.date.greater(datetime.now() - timedelta(days=5)),
    gx.some(
        Message.sender.equals(john),
        Message.reciever.equals(john),
        Message.reciever.is_not(jane)
    )).all()
```

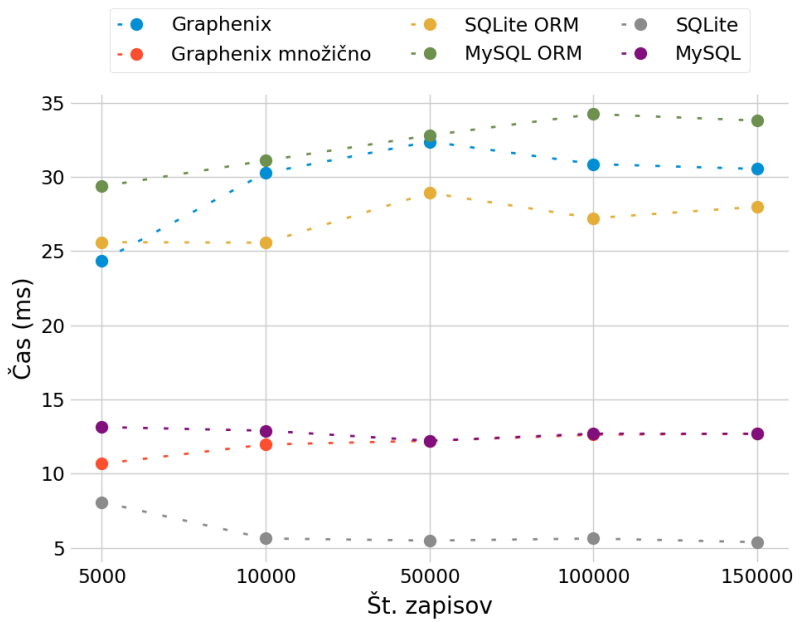
Vgnezdne poizvedbe

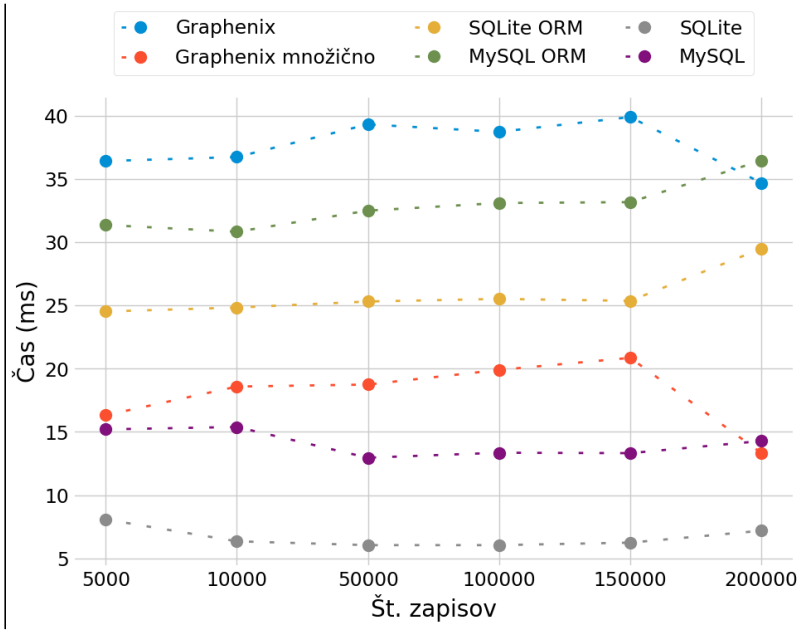
Nabor uporabnikov, njihovih nalog in prejetih sporočil, kjer na sporočila vežemo še pošiljatelja

```
_, view = User.link(
    tasks=Task.order(Task.content).limit(3),
    recieved=Message.link(sender=User).limit(5)
).filter(User.name.iregex('john.*')).all()
```

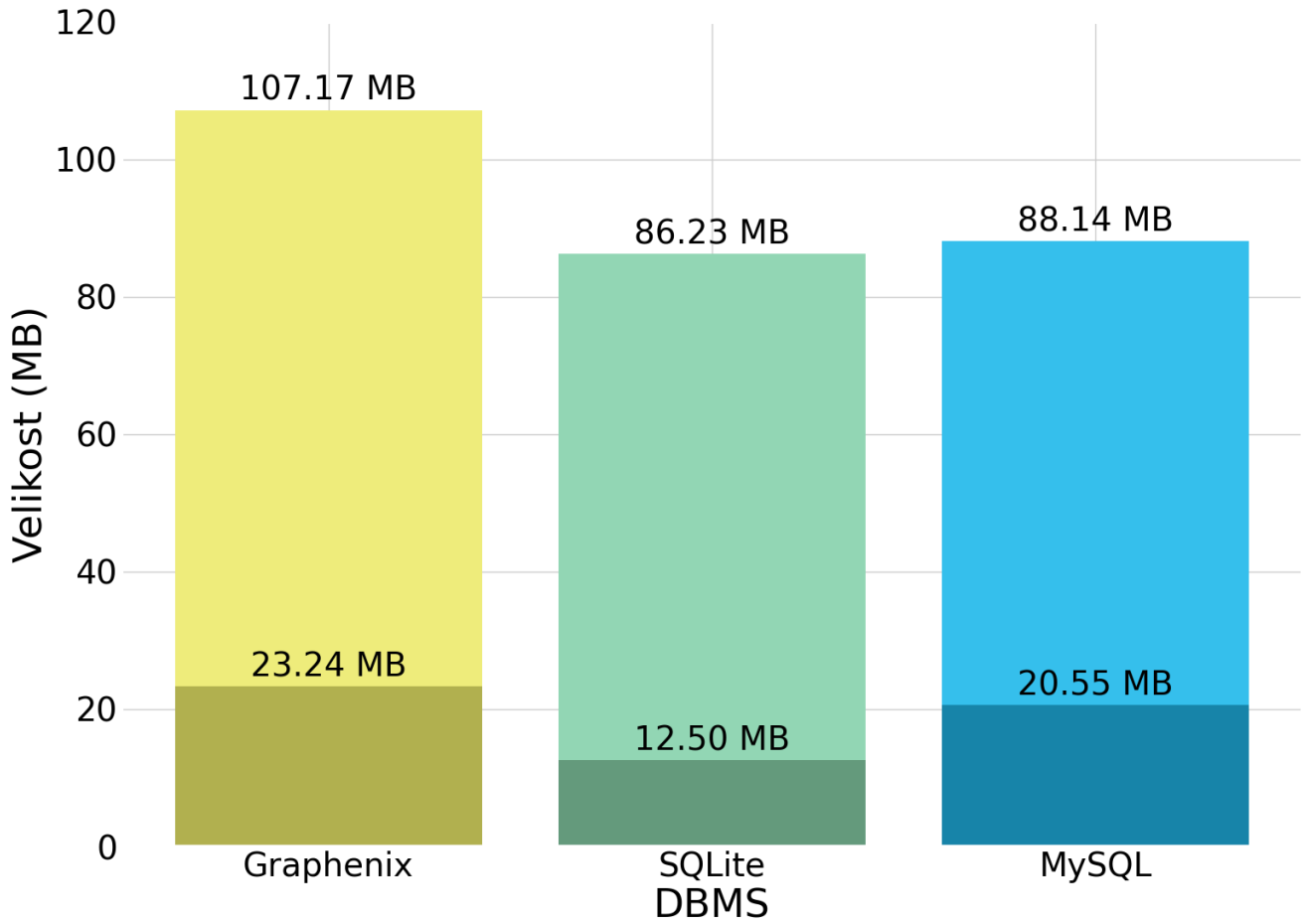
```
{
    "name": "John Doe",
    "tasks": [
        {"content": "Finish the diploma"},
    ],
    "recieved": [
        {"content": "Hello", "sender": {}},
    ]
}
```

Vstavljanje zapisov (brez in z indeksiranim atributom)

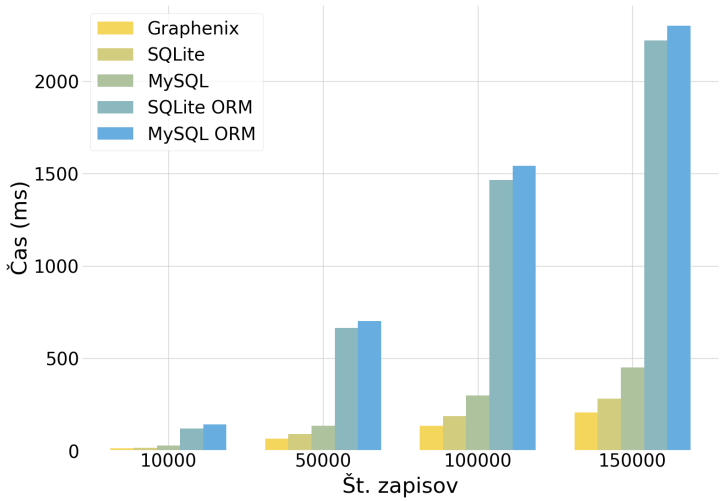


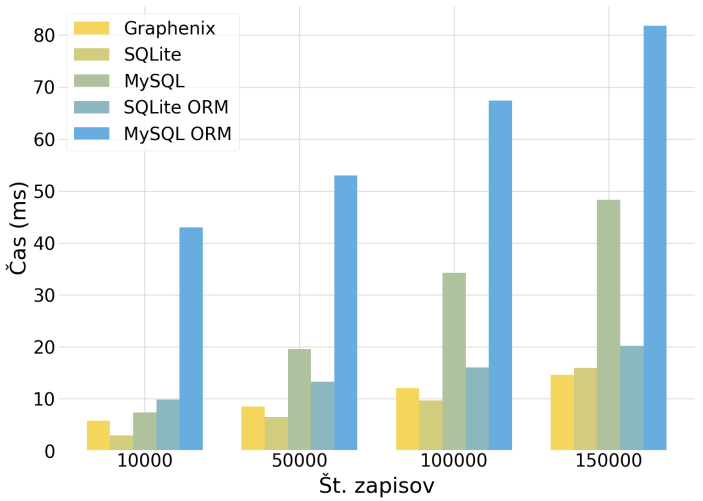


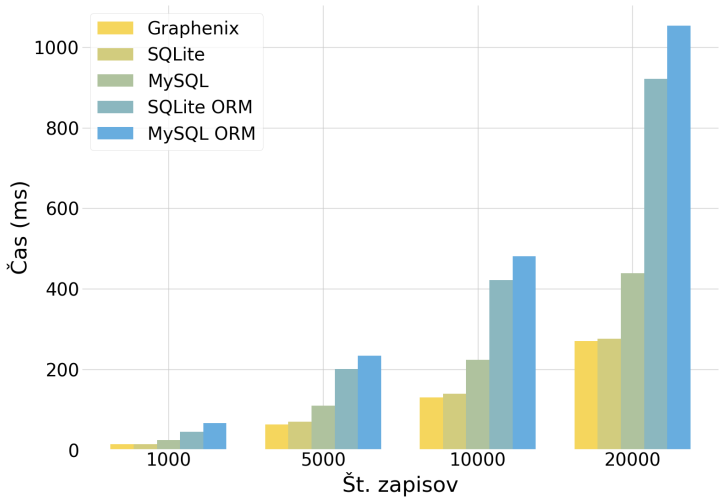
Velikost podatkovne baze na disku

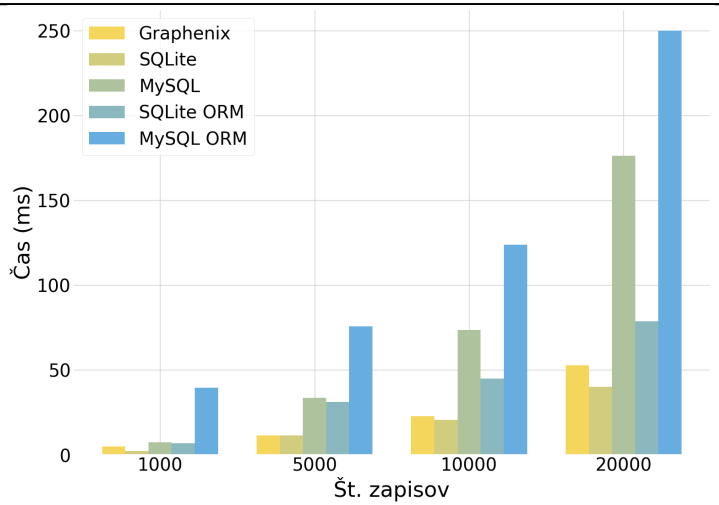


Brez omejitev, s filtrom, s povezavami, agregacijsko

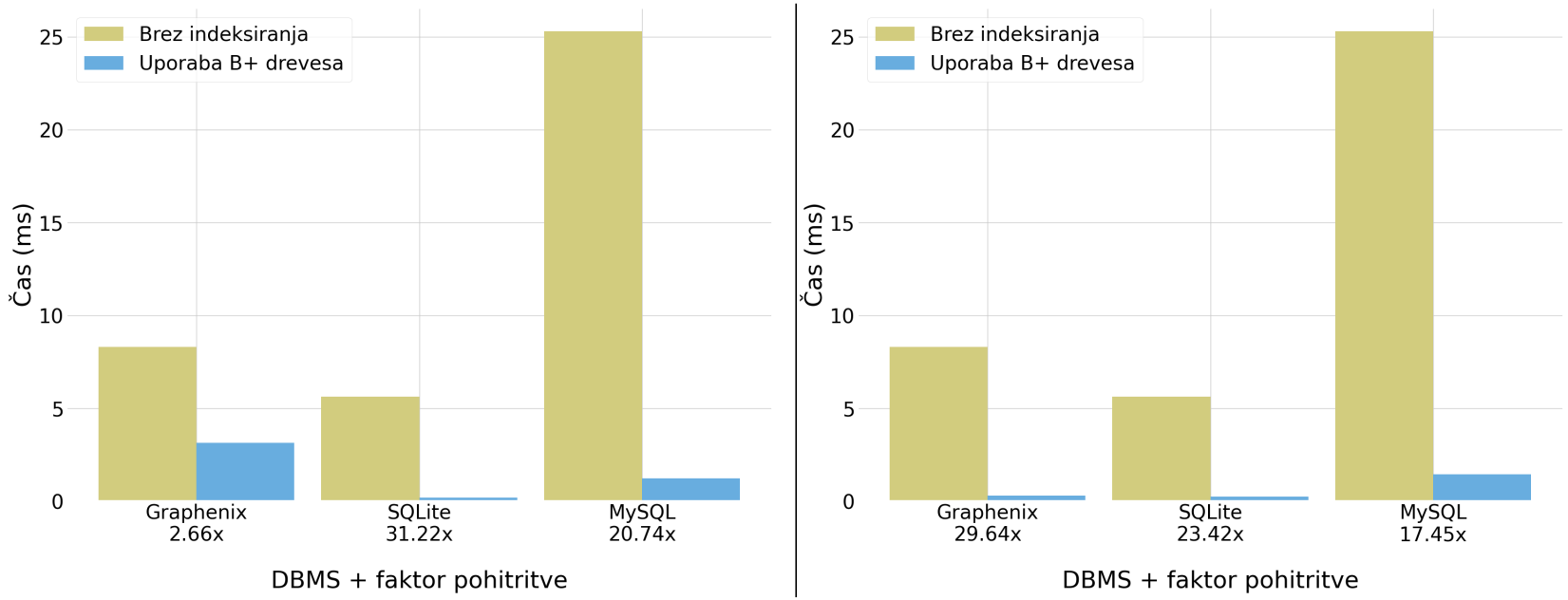




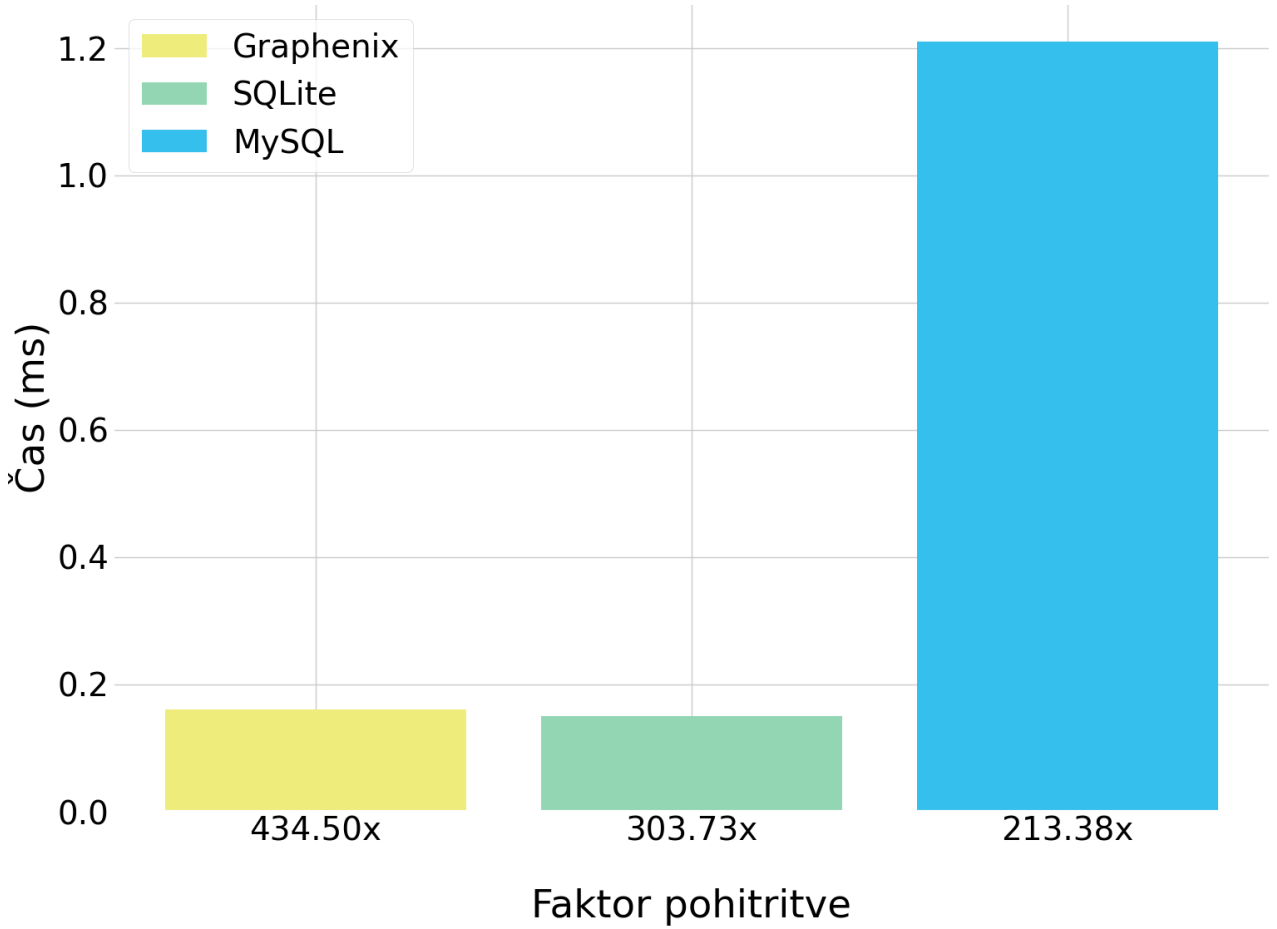




Indeksiranje pred in po optimizaciji (število zapisov = 10^5)



Indeksiranje nad večjo bazo podatkov (10^6 zapisov)



Sklepne ugotovitve

- Kje je rešitev uporabna?
- Kaj rešitvi manjka za uporabo v produkcijskem okolju?
- Ali je bil razvoj uspešen?

Hvala za pozornost