

Razvoj SUPB z integracijo v programski jezik Python

Janez Sedeljšak

Mentor: doc. dr. Boštjan Slivnik

Somentor: asist. dr. Marko Požnenel

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

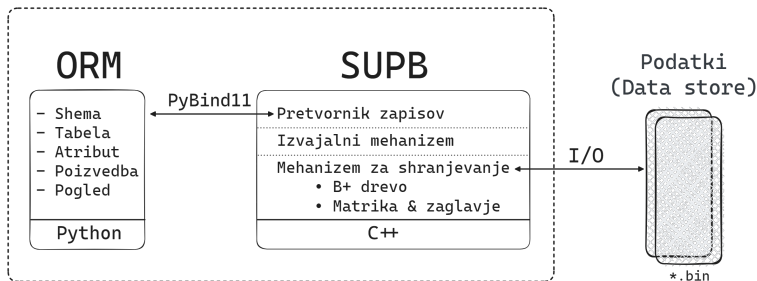
js0578@student.uni-lj.si

September 11, 2023

Motivacija

- **Spoznati delovanje podatkovnih baz**
 - Izogib uporabi anti-vzorcev
 - Boljša implementacija podatkovnega sloja
- **Razvoj lastne SUPB rešitve**
 - SUPB na nivoju programskega jezika C++
 - Indeksiranje z uporabo B+ dreves
 - Intuitiven način komunikacije s podatkovno bazo
 - Izhodišče sta MySQL in SQLite

Arhitektura rešitve Graphenix



Struktura shranjenih podatkov

Zaglavje

Prosto mesto: 2.	
ID	VRSTICA V MATRIKI
0	-1 (pobrisano)
1	0
2	1
3	-1 (pobrisano)
4	3

Matrika podatkov

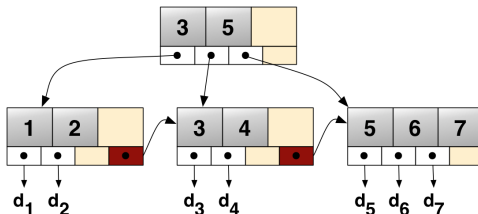
ODMIK	VRSTICA V MATRIKI
0	zapis (ID = 1)
1	zapis (ID = 2)
2	prazno (izbrisan zapis)
3	zapis (ID = 4)
...	

Zapisi

Atributi

Indeksiranje z uporabo B+ drevesa

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



Definiranje sheme

```
class User(gx.Model):  
    name =      gx.Field.String(size=100)  
    ident =     gx.Field.Int().as_index()  
    tasks =     gx.Field.VirtualLink("user")  
    sent =      gx.Field.VirtualLink("sender")  
    received =  gx.Field.VirtualLink("receiver")
```

```
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()  
    receiver =   gx.Field.Link()
```

Poizvedovanje

```
# uporabniki urejeno po imenu - padajoče
_, view = User.order(User.name.desc()).all()

# uporabniki in njihove naloge
_, view = User.link(tasks=Task).all()

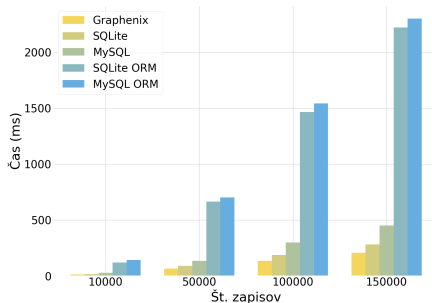
# število sporočil in datum zadnjega sporočila po uporabnikih
counts = Message.agg(by=Message.sender,
    count=gx.AGG.count(), latest=gx.AGG.max(Message.date))
```

Vgnezdene poizvedbe

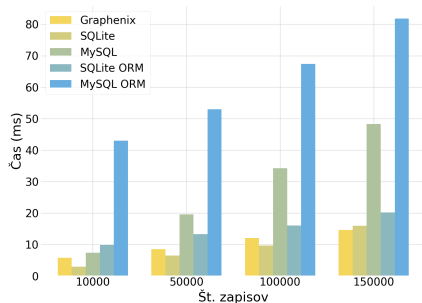
Nabor uporabnikov, njihovih nalog in prejetih sporočil, kjer na sporočila vežemo še pošiljatelja

```
_, view = User\  
  .link(  
    tasks=Task.order(Task.content).limit(3),  
    received=Message.link(sender=User)  
  )\  
  .filter(User.name.iregex('john.*'))\  
  .all()  
  
{  
  "name": "John Doe",  
  "tasks": [  
    {"content": "Finish the diploma"},  
  ],  
  "received": [  
    {"content": "Hello", "sender": {}},  
  ]  
}
```


Poizvedovanje

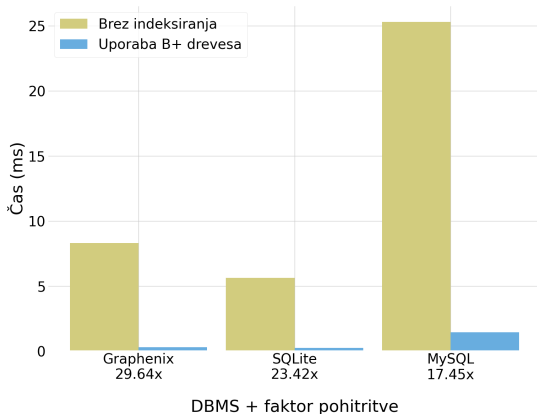


SELECT * FROM users



SELECT * FROM users
WHERE is_admin = 1
ORDER BY ident
LIMIT 500

Iskanje s pomočjo indeksiranja (10^5 zapisov)



SELECT * FROM users WHERE ident = 5432

Sklepne ugotovitve

- Kje je rešitev uporabna?
- Kaj rešitvi manjka za uporabo v produkcijskem okolju?

Hvala za pozornost