Databázy: Prehľad a porovnanie rôznych noSQL-databáz

Bc. Alexander Sárközy

Bc. Ladislav Rajcsányi

Bc. Tomáš Kukumberg

Bc. Maksim Mištec

Databázy ktoré sme použili:

- MongoDB
- Redis
- CouchDB
- Neo4j

MongoDB

- Nerelačná dokumentová databáza
- JSON-like storage
- SEGA ju používa na správu herných účtov
- Aer Lingus ju používa na správu leteniek
- Sourceforge ju používa na správu údajov

```python
def __init__(self, statistics: Statistics) -> None:
    """Initializes the MongoDbDAO Class.

    Args:
        statistics (Statistics): Database Testing Statistics Object.
    """
    super().__init__()

    self.__database_type: str = 'MongoDB'
    self.__port: int = 27017
    self.__connection: MongoClient = self.create_connection(
        username=os.getenv('ASOS_USERNAME'),
        password=os.getenv('PASSWORD')
    )
    self.__statistics: Statistics = statistics

def create_connection(self, **kwargs: str) -> MongoClient:
    """Creates new MongoDB Connection.

    Args:
        **kwargs (str): Keyword Arguments ('username' and 'password'
        expected).

    Returns:
        MongoClient: New MongoDB Connection.
    """
    try:
        connection = MongoClient(
            host=f"mongodb://localhost:{self.__port}/",
        )

    except errors.ServerSelectionTimeoutError as err:
        connection = None
        print('pymongo ERROR:', err)
        exit(-1)

    return connection
```

```python
def read_data(self, **kwargs: str) -> None:
    """Reads every entry in Collection.

    Args:
        **kwargs (str): Keyword Arguments ('database' and 'collection'
        expected).
    """
    collection = self.__connection[kwargs['database']
                                   ][kwargs['collection']]

    start_time = time.time()

    for entry in collection.find():
        print(entry)

    self.__statistics.add_execution_time(
        database_type=self.__database_type,
        database=kwargs['database'],
        dataset=kwargs['collection'],
        action='read',
        time=time.time() - start_time
    )
```

```python
def insert_data(self, **kwargs: Union[str, List[dict]]) -> None:
    """Inserts entries to Collection.

    Args:
        **kwargs (Union[str, List[dict]]): Keyword Arguments ('database',
        'collection' and 'data' expected).
    """
    try:
        start_time = time.time()
        (
            self.__connection[kwargs['database']][kwargs['collection']]
            .insert_many(kwargs['data'])
        )
        self.__statistics.add_execution_time(
            database_type=self.__database_type,
            database=kwargs['database'],
            dataset=kwargs['collection'],
            action='insert',
            time=time.time() - start_time
        )

    except errors.BulkWriteError as bwe:
        print(bwe.details)
        raise
```

```python
def update_data(self, **kwargs: Union[str, List[dict]]) -> None:
    """Updates entries in Collection.

    Args:
        **kwargs (Union[str, List[dict]]): Keyword Arguments ('database',
        'collection', 'old_values' and 'new_values' expected).
    """
    start_time = time.time()
    (
        self.__connection[kwargs['database']][kwargs['collection']]
        .update_many(
            kwargs['old_values'],
            kwargs['new_values']
        )
    )
    self.__statistics.add_execution_time(
        database_type=self.__database_type,
        database=kwargs['database'],
        dataset=kwargs['collection'],
        action='update',
        time=time.time() - start_time
    )

def delete_data(self, **kwargs: str) -> None:
    """Removes every entry from Collection.

    Args:
        **kwargs (str): Keyword Arguments ('database' and 'collection'
        expected).
    """
    start_time = time.time()
    (
        self.__connection[kwargs['database']][kwargs['collection']]
        .delete_many({})
    )
    self.__statistics.add_execution_time(
        database_type=self.__database_type,
        database=kwargs['database'],
        dataset=kwargs['collection'],
        action='delete',
        time=time.time() - start_time
    )
```

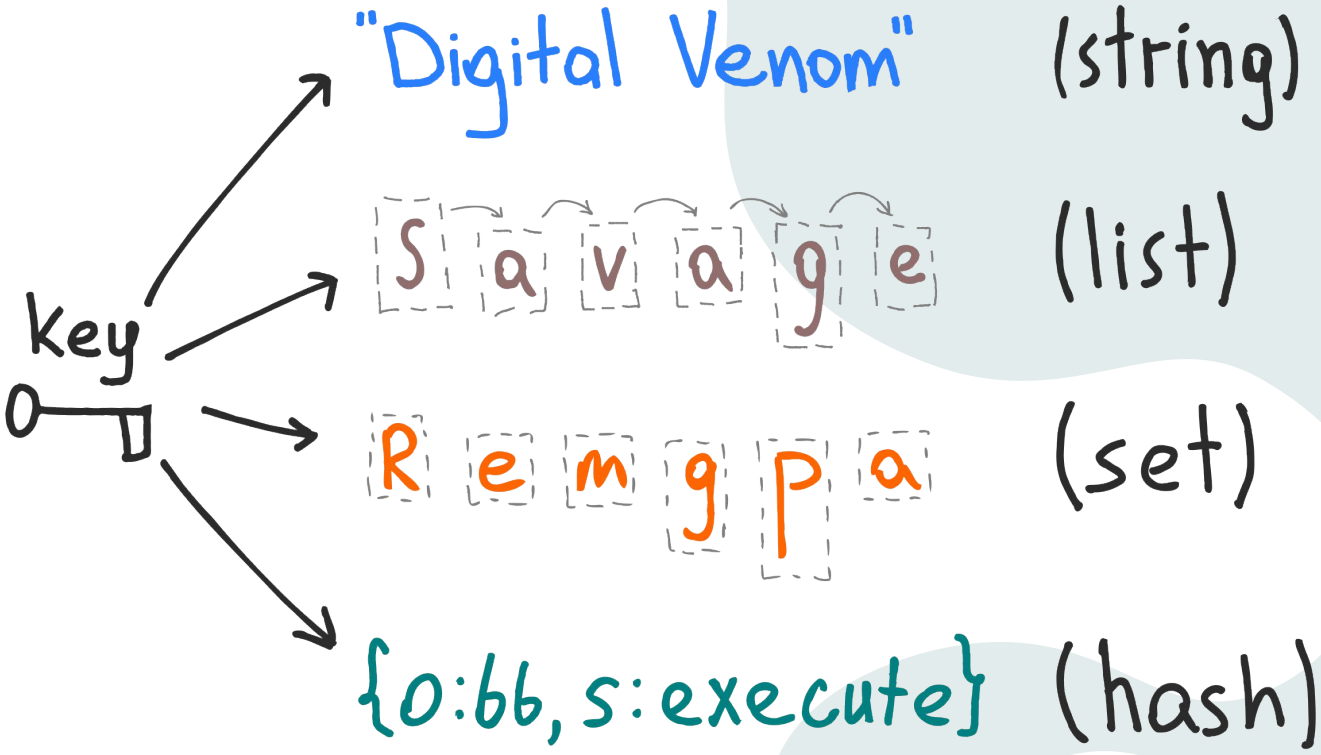
MongoDB data príklad

```
{
   "student": {
      "name": "John",
      "class": "Intermediate",
      "address": {
         "street": "2293 Example Street",
         "City": "Chicago",
         "State": "IL"
      }
   }
}
```

Redis

- Údaje typu kľúč-hodnota
- Ideálne na ukladanie informácií
- Pinterest ju používa na ukladanie zoznamov obrázkov a galérií
- Coinbase ju používa na autorizáciu kurzových bodov
- Twitter ju používa na správu časovej osi

Redis data príklad



CouchDB

- Nerelačná dokumentová databáza
- Jednoduchá horizontálna škálovateľnosť na rôznych zariadeniach
- Spoločnosť United Airlines používa CouchDB pre zábavné systémy počas letu vo viac ako 3 000 lietadlách
- Používa ju BBC pre dynamickú CMS platformu

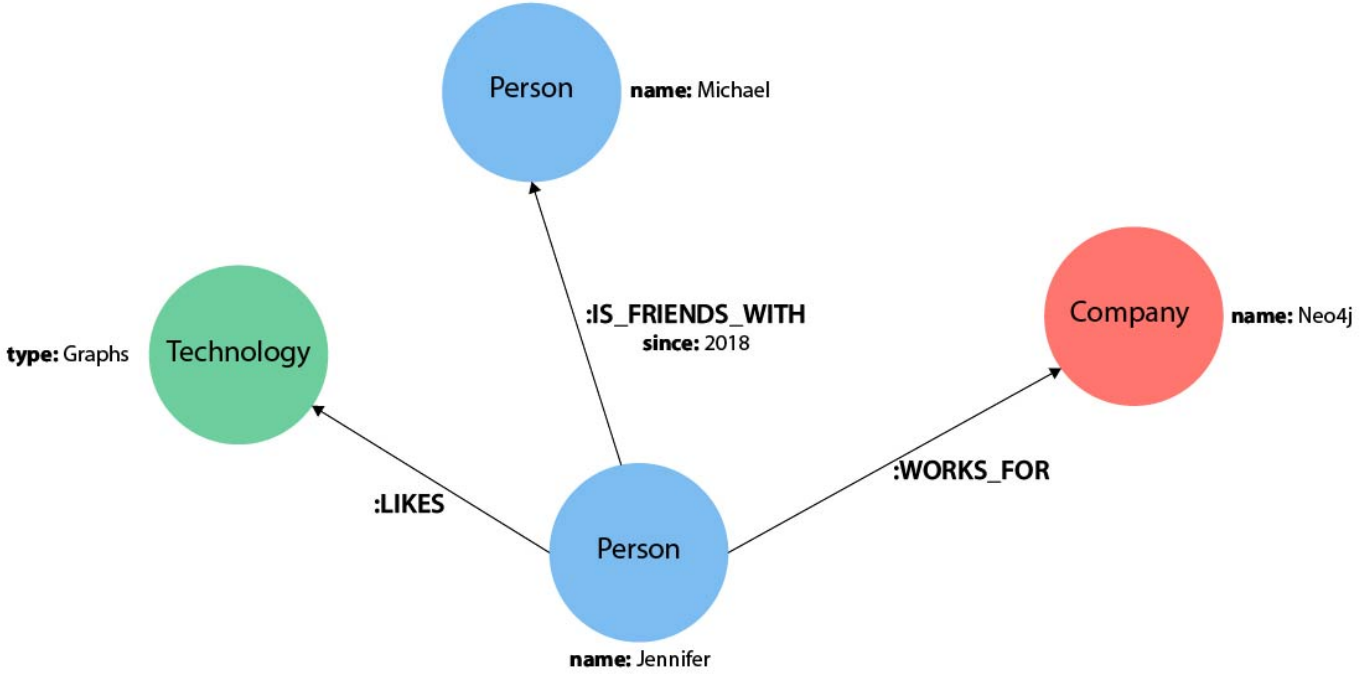
CouchDB data príklad

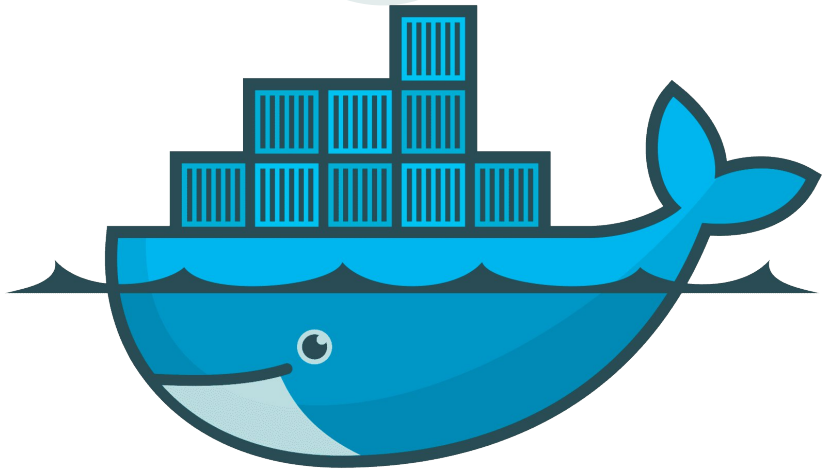

```
{
 "id": "c4e8630bfa328d3132965bd7cd001dd1",
 "key": "c4e8630bfa328d3132965bd7cd001dd1",
 "value": {
  "rev": "2-7c65286b473f46fbb134ddd87feb0d4f"
 },
 "doc": {
  "_id": "c4e8630bfa328d3132965bd7cd001dd1",
  "_rev": "2-7c65286b473f46fbb134ddd87feb0d4f",
  "tutorial": "CouchDB Tutorial",
  "category": "Databases",
  "number_of_topics": 7
 }
}
```

Neo4j

- Grafové databázy
- Rýchlejší výkon dotazov ako relačné databázy
- Cisco ju používa na analýzu problémov zákazníckej podpory s cieľom predvídať chyby
- Walmart ju používa na poskytovanie relevantných propagácií a odporúčaní produktov.

Neo4j data príklad

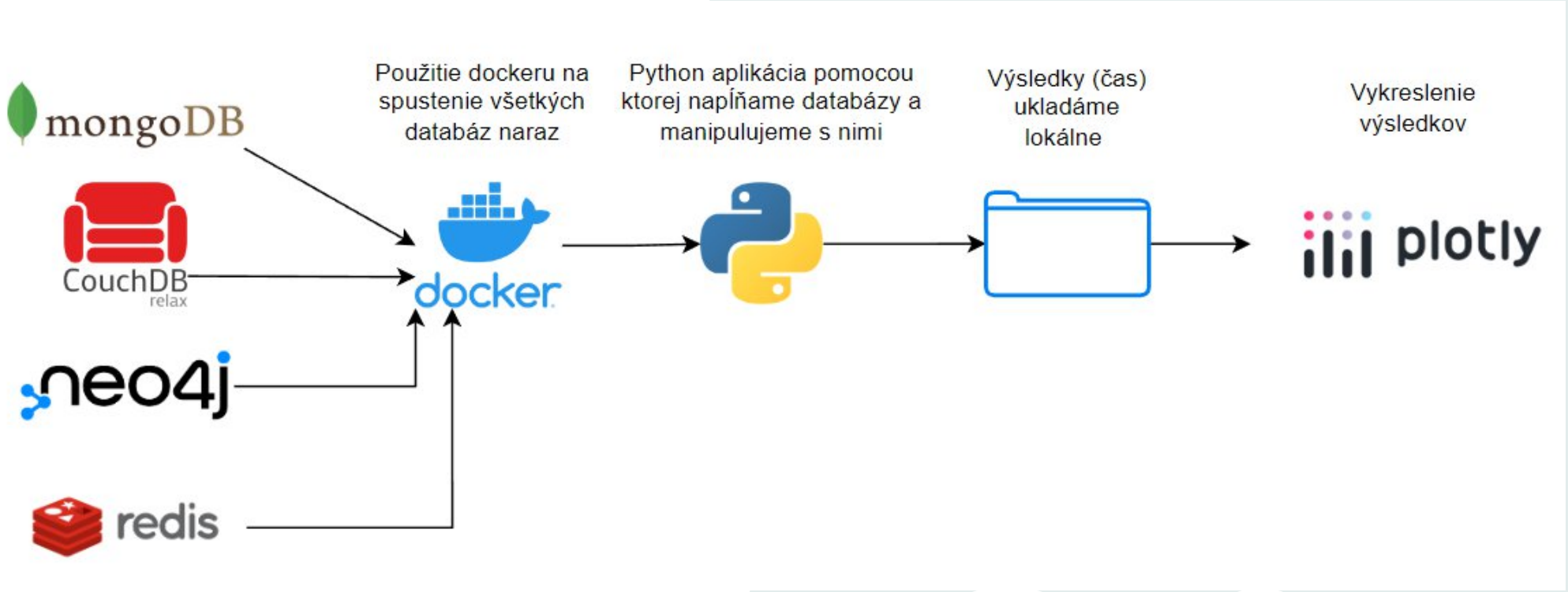




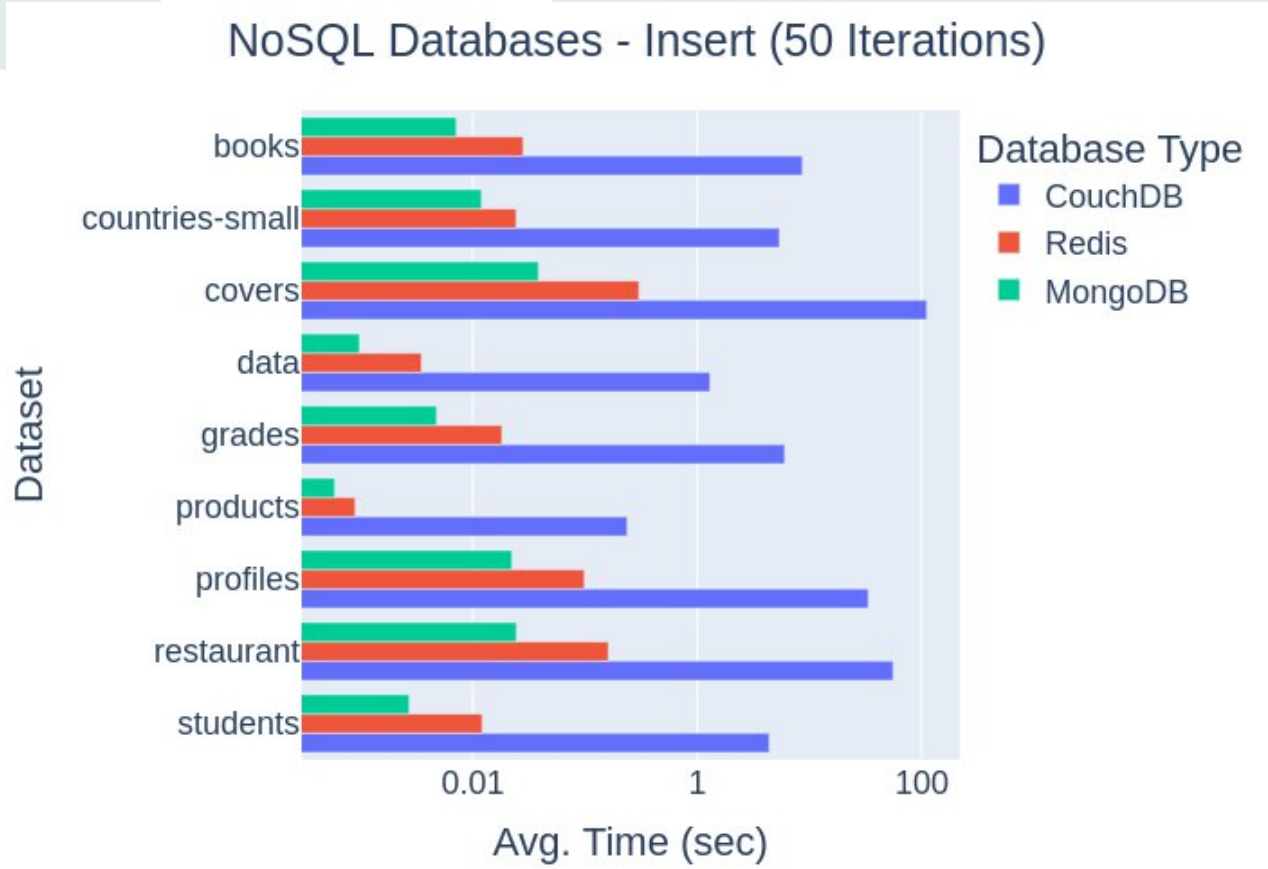
docker

Python virtual environment

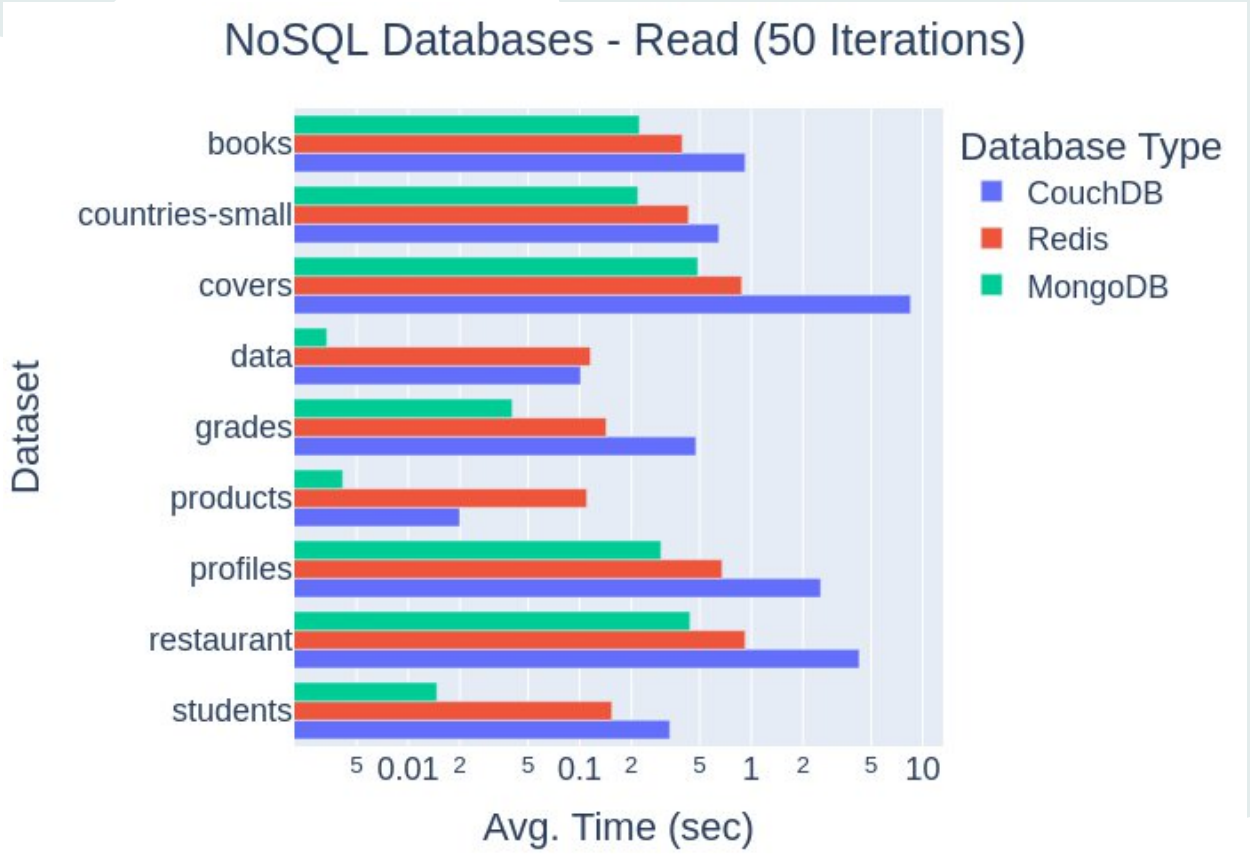
Implementácia



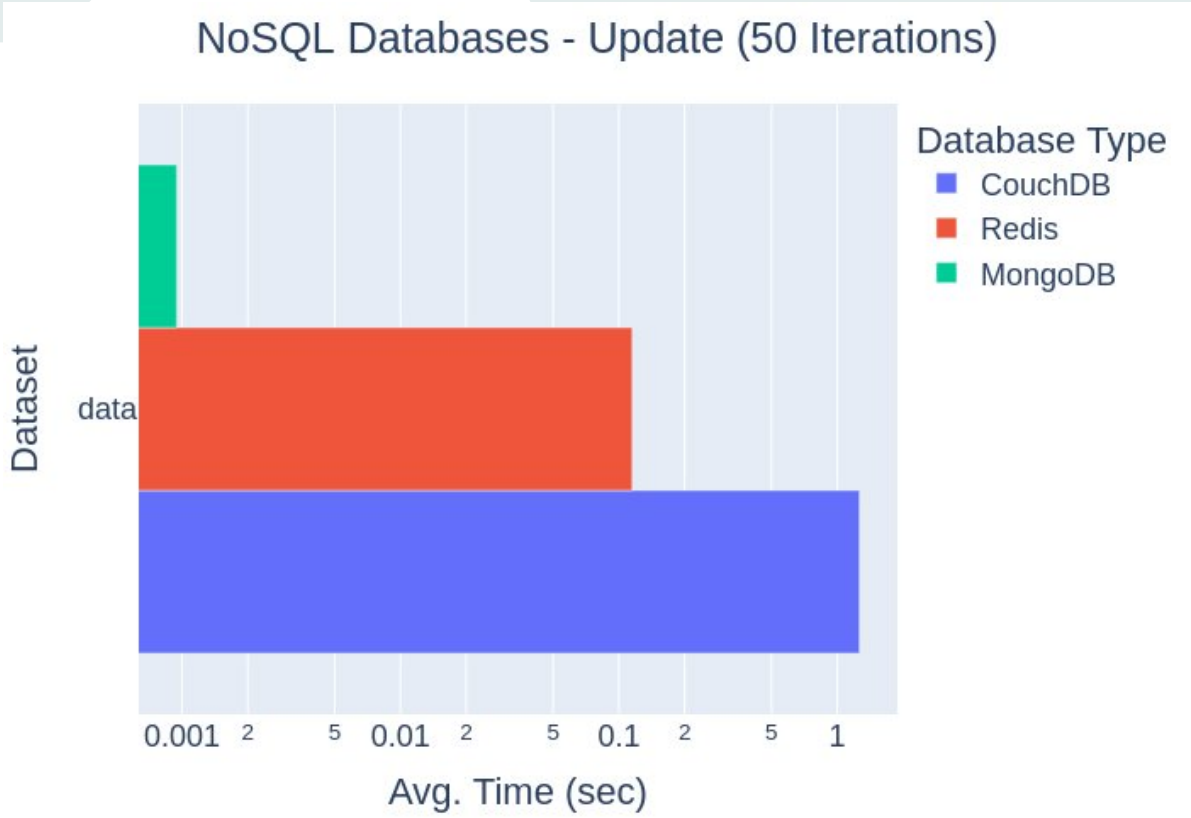
Výsledky - INSERT



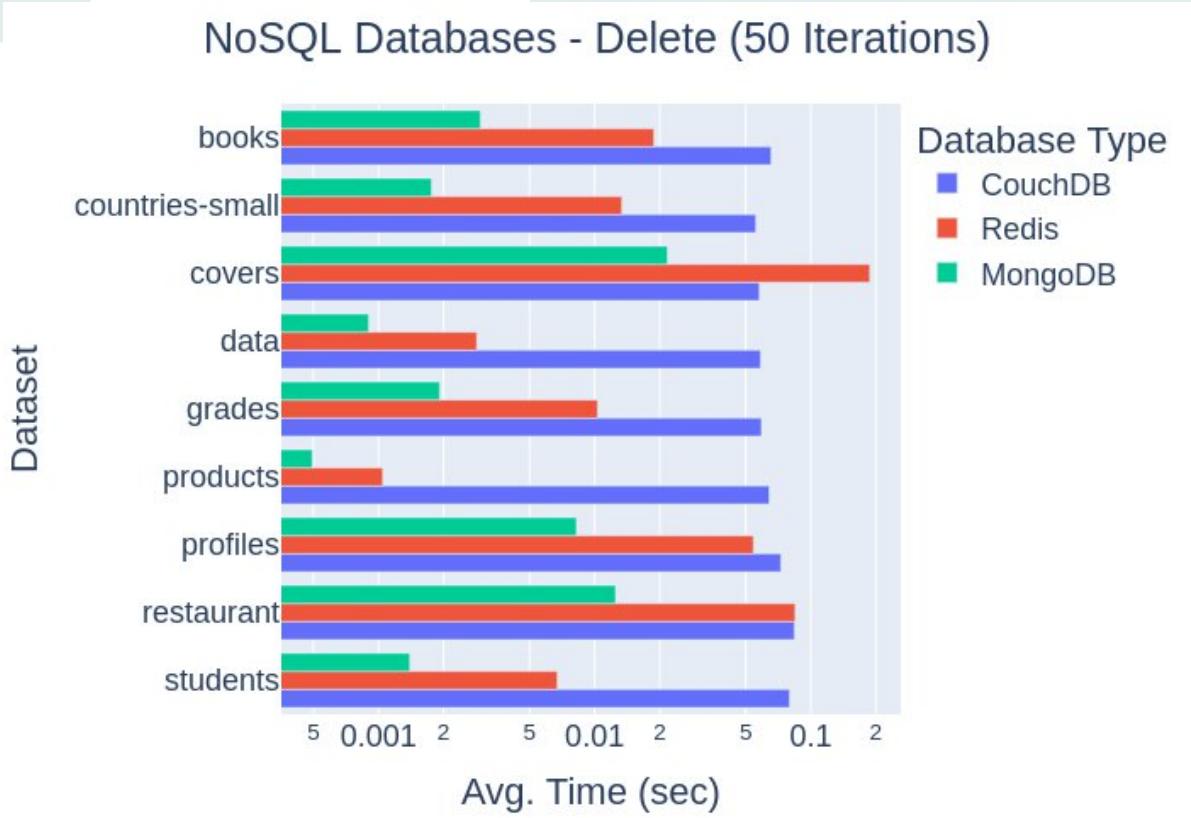
Výsledky - READ



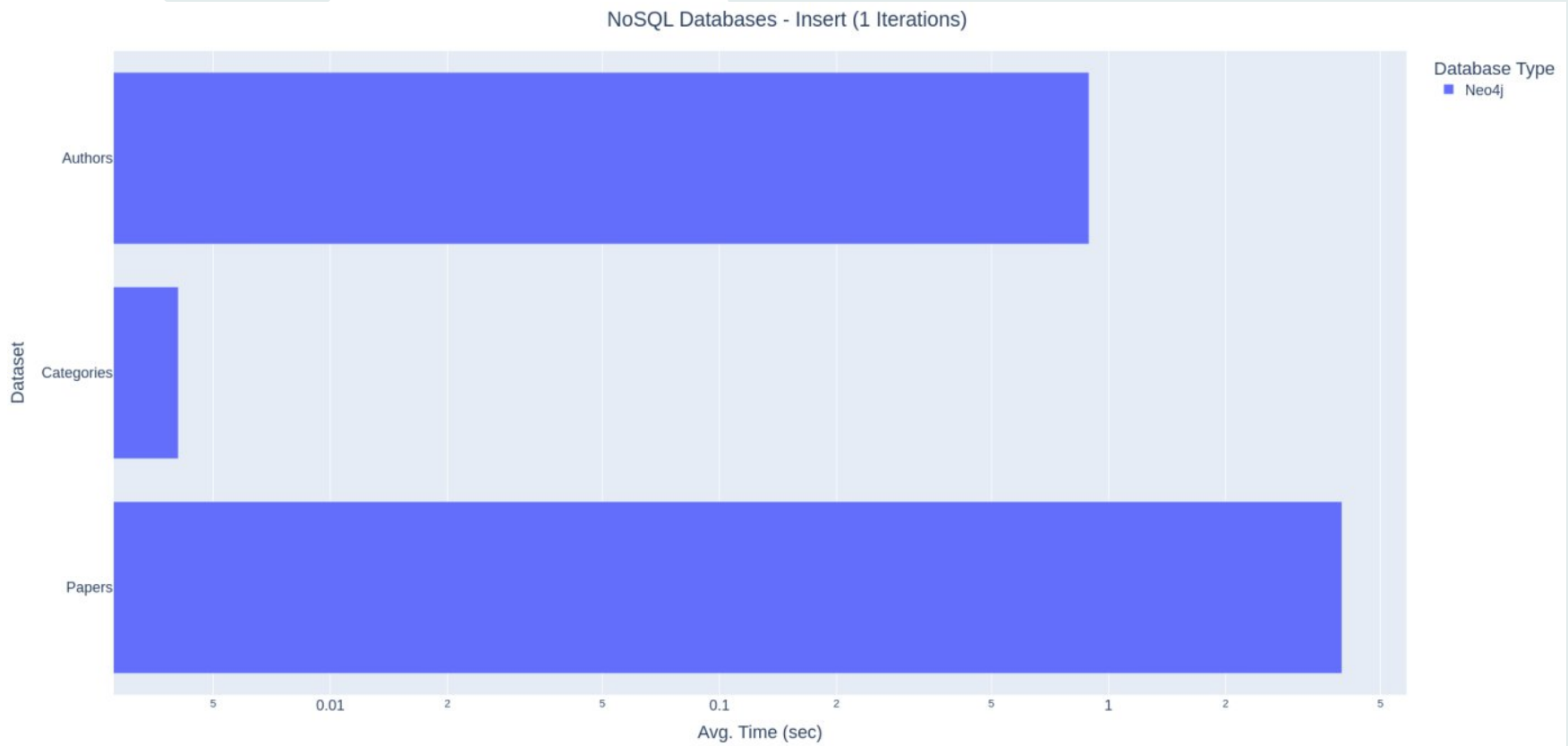
Výsledky - UPDATE



Výsledky - DELETE



Neo4j výsledky



Ďakujeme za pozornosť

Otázka #1 na skúšku

Ktorá z nasledujúcich databáz má grafovú štruktúru?

- Oracle
- Redis
+ Neo4j
- MongoDB
- CouchDB
- Cassandra

Otázka #2 na skúšku

Aké 4 hlavné typy noSQL databáz poznáme? (Vymenujte)

Správna odpoveď:

Dokumentové databázy, Kľúč-Hodnota databázy, Stĺpcovo orientované databázy a Grafové databázy

Alebo

Document databases, Key-value databases, Column-oriented databases a Graph databases

Zdroje

https://neo4j.com/developer/python/

https://www.mongodb.com/languages/python https://realpython.com/python-redis/

https://stackoverflow.com/questions/62804624/run-cassandra-cqlsh-with-python-3-on-windows-10

https://www.mongodb.com/compatibility/json-to-mongodb

https://en.wikipedia.org/wiki/Document-oriented_database

https://docs.datastax.com/en/developer/python-driver/3.25/

https://pypi.org/project/cassandra-driver/