ZTBD – projekt
Temat: Analiza porównawcza wydajności baz danych
Data: 28.01.2023
Zespół: Anna Dybel, nr. albumu 148901 Krzysztof Dulęba, nr. albumu 148900

1. Cel projektu

Celem projektu było wybranie źródła danych, pobranie danych i umieszczenie ich w trzech wybranych bazach danych (jedna o modelu relacyjnym i dwie o modelu nierelacyjnym), a następnie przeprowadzenie analizy porównawczej wydajności baz na podstawie przeprowadzonych testów.

2. Wykorzystane technologie

1. Git – system kontroli wersji
2. Python v.3.9.0 – język programowania, w którym został opracowany projekt
3. Beautiful Soup – biblioteka umożliwiająca scrapowanie danych z witryn internetowych
4. FastApi v.0.88.0 – biblioteka w Python służąca do tworzenia API
5. SQLAlchemy v.1.4.45 – Bilbioteka Python do zarządzania bazą PostgreSQL
6. PyMongo v.4.3.3 – Biblioteka Python do zarządzania bazą MongoDB
7. Redis-Py 4.4.2 - oprogramowanie klienckie do pythona do obsługi Redisa
8. Docker – program służący do wirtualizacji na poziomie systemu operacyjnego
9. ReactJS – biblioteka języka programowania JavaScript, która wykorzystywana jest do tworzenia interfejsów graficznych aplikacji internetowych.

3. Konfiguracja

1. Wymagania wstępne: instalacja lub aktualizacja git, python, pip, docker i nodejs
2. Sklonowanie repozytorium: https://github.com/Anna21Tori/ZTBD.git
3. Dalsze postępowanie według zamieszczonego pliku README.md

4. Dane

Tematyka danych: Książki

Źródło danych: www.lubimyczytac.pl

Dane zostały pobrane z ww. strony za pomocą web scrapera napisanego w języku Python z użyciem biblioteki Beautiful Soup. Dane zajmują około 1.5GB.

Dane składają się z następujących kolumn:

1. Title – tytuł książki
2. Author – imię (lub imiona) i nazwisko autora
3. Categories – kategorie książki
4. Publishing – wydawca
5. ISBN – międzynarodowy znormalizowany numer książki
6. Description – opis książki
7. OriginalTitle – tytuł książki w oryginalnym języku
8. Date – data wydania książki (świat)
9. DatePol – data wydania książki w Polsce
10. Pages – ilość stron książki
11. Lang – język książki
12. Translator – tłumacz książki
13. Comments – komentarze od użytkowników do książki
14. Quotes – cytaty z książki

5. Obiekty badań

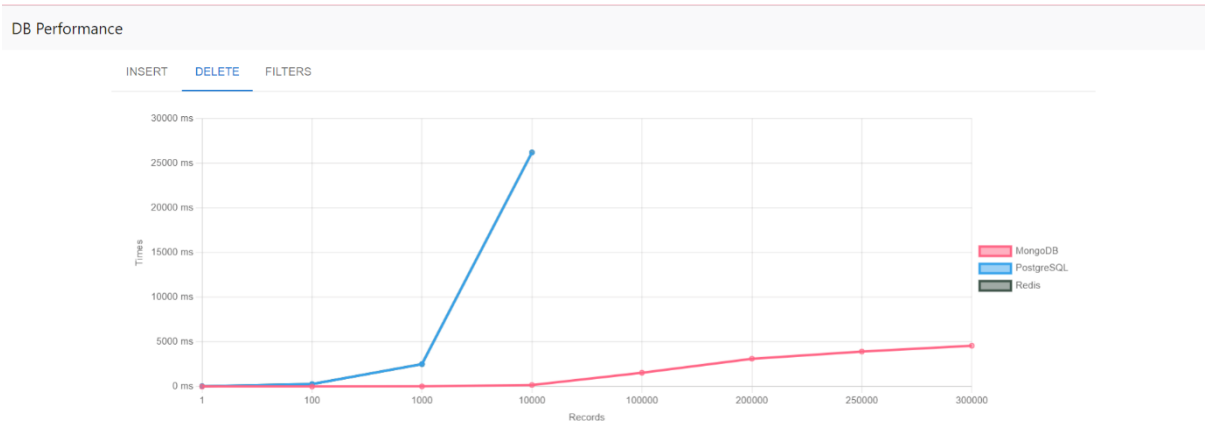
Wybrane przez nas bazy danych to:

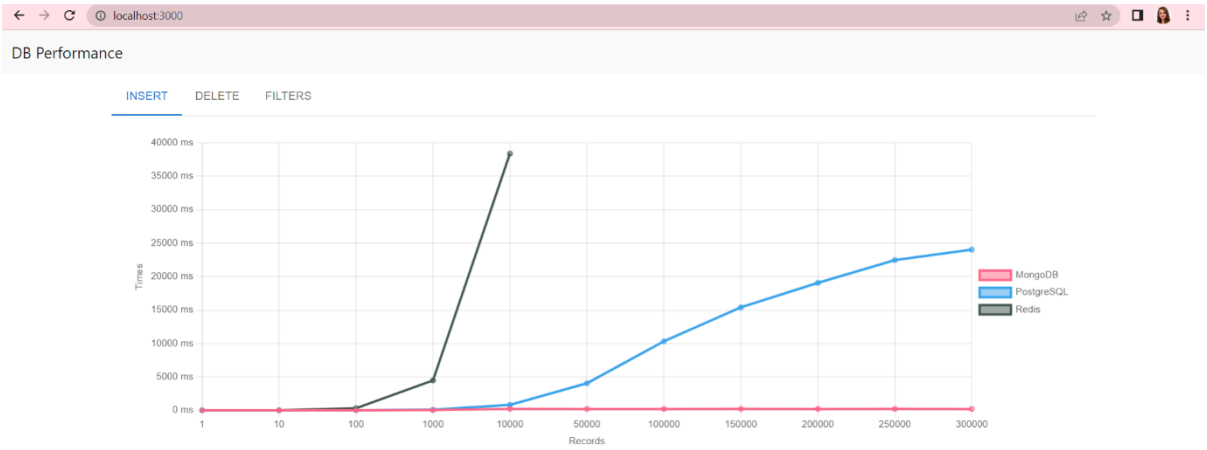
- PostgreSQL – system zarządzania relacyjnymi bazami danych
- MongoDB – system zarządzania nierelacyjnymi bazami danych, w którym dane są przechowywane jako dokumenty w formacie JSON.
- Redis – system zarządzania nierelacyjnymi bazami danych, w którym dane są przechowywane w strukturze klucz-wartość w pamięci operacyjnej serwera.

6. GUI

Po poprawnym uruchomieniu API (podgląd http://127.0.0.1:8000/docs) oraz włączeniu aplikacji webowej użytkownika ma do dyspozycji trzy zakładki:

- INSERT – wynik badania operacji dodawania w postaci wykresów liniowych
- DELETE – wynik badania operacji usuwania w postaci wykresów liniowych
- FILTERS – wyniki badania operacji wybierania danych w postaci wykresów słupkowych





7. Metoda badań

Porównanie wydajności relacyjnych i nierelacyjnych baz danych polegało na przeprowadzeniu operacji dodania, usuwania i wybierania danych. Badanie operacji dodawania zostało powtórzone 3 razy dla 1, 10, 100, 1000, 10000, 50000, 100000, 150000, 200000, 250000 i 300000 rekordów, a następnie obliczony zostały czas średni. Badanie operacji usuwania zostało powtórzone 3 razy. Dla bazy MongoDB dla 1, 10, 100, 1000 rekordów, natomiast dla bazy PostgreSQL dla 1, 10, 100, 1000, 10000, 50000, 100000, 150000, 200000, 250000 i 300000 rekordów. Badanie operacji wybierania zostało powtórzone 10 razy.

8. Opis wykonanych prac

1. Utworzenie skryptu w języku Python z wykorzystaniem biblioteki Beautiful Soup, którego zadaniem było pobranie danych ze strony www.lubimyczytac.pl poprzez wykorzystanie techniki web scraping. Dane zostały zapisane w plikach CSV.
2. Utworzenie skryptu loader_data służącego do załadowania danych z plików CSV w formacie umożliwiającym wykorzystanie ich w dalszej części projektu.

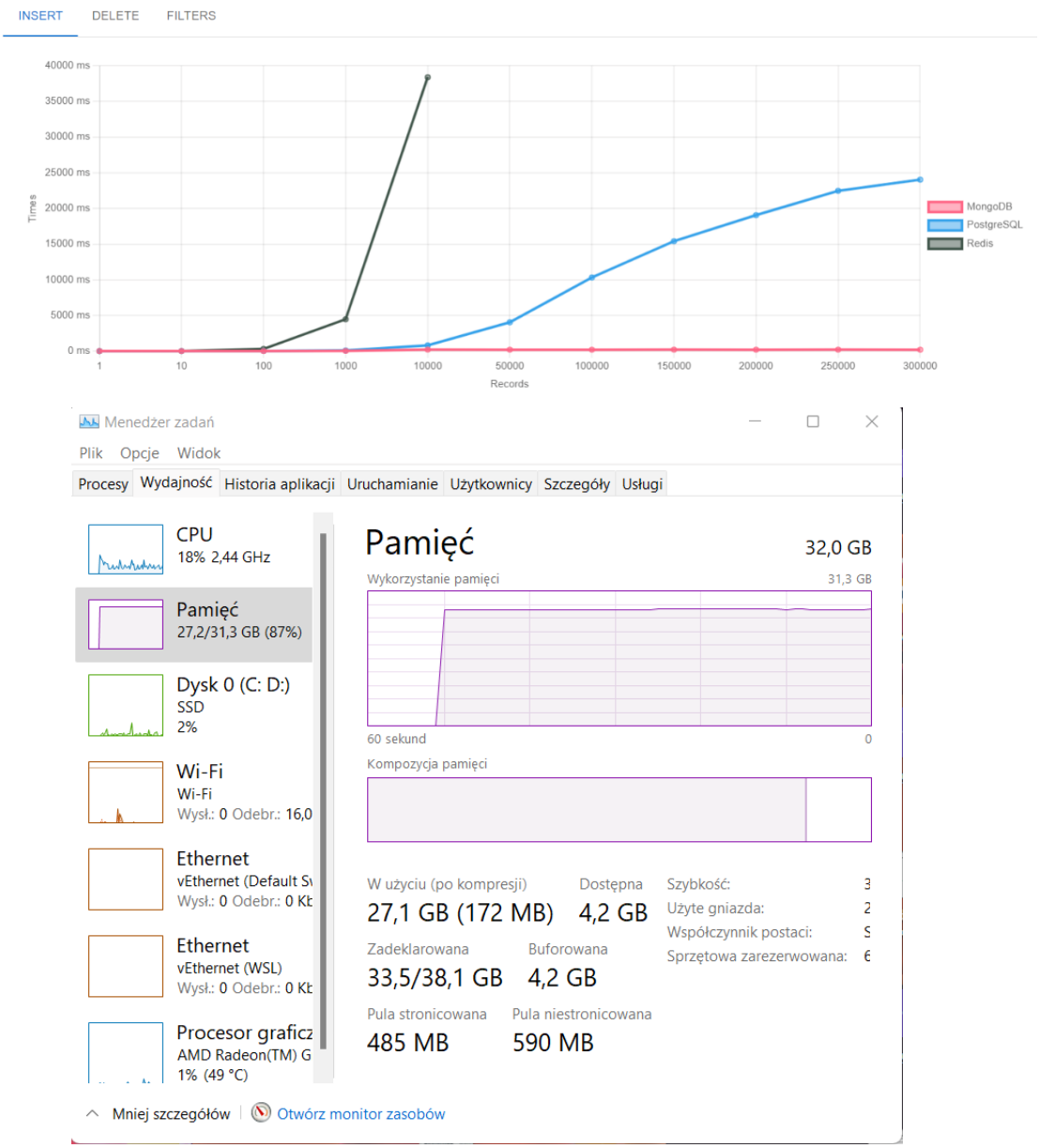
3. Skonfigurowanie obrazów baz danych wybranych do przetestowania w niniejszym projekcie.
4. Stworzenie modeli i metod pozwalających na zarządzanie bazami danych poprzez dodawanie, usuwanie i wybieranie rekordów.
5. Stworzenie testów mających na celu przeprowadzenie pomiarów czasowych operacji i zapis otrzymanych wyniku do plików w formacie json.
6. Dodanie API z wykorzystaniem FastAPI w celu udostępnienia wyników badań aplikacji webowej.
7. Utworzenie aplikacji webowej wykorzystującej wyniki badań w celu ich prezentacji.
8. Podsumowanie otrzymanych wyników badań.

9. Wyniki badań

1. Operacja dodawania rekordów

   Wstawianie danych do systemów bazodanowych jest jedną z najczęściej wykonywanych operacji, a więc wydajność jej wykonania jest dość istotna przy wyborze bazy.

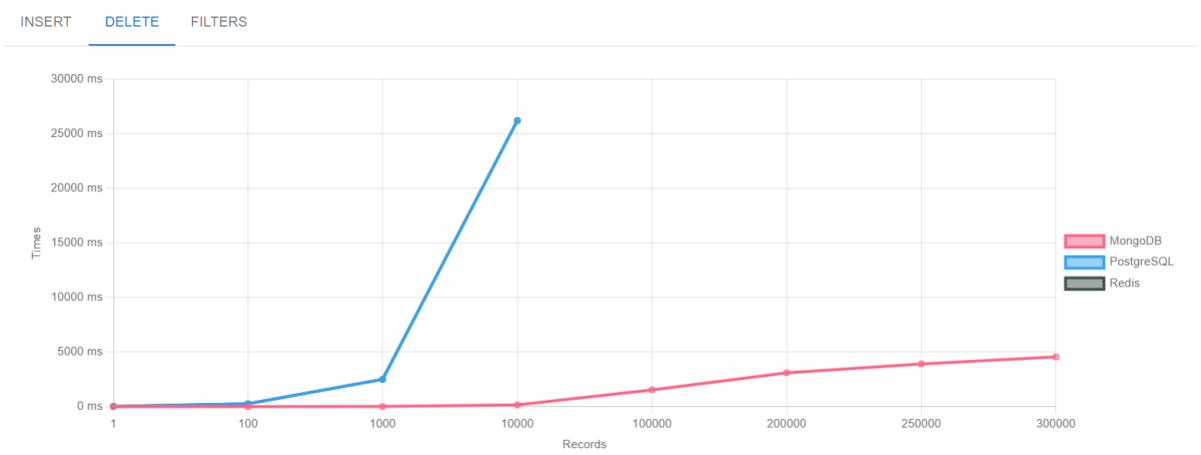
   Na poniższym wykresie zostały przedstawione wyniki przeprowadzonego badania operacji dodawania rekordów do bazy PostgreSQL, MongoDB oraz Redis. Jak można zauważyć początkowo przy małej liczby rekordów wszystkie bazy szybko w miarę podobnych czasie wykonywały operacje dodawania. Przy dodawaniu 1000 rekordów można jednak zauważyć spadek wydajności pomiędzy MongoDB i PostgreSQL. Przy maksymalnej liczbie rekordów dodawanie ich do bazy MongoDB jest prawie dwa razy szybsze niż do bazy PostgreSQL. Badanie wydajności operacji dodawania do bazy Redis zostało przerwane po próbie dodania 10000 rekordów ze względu na to, że przed dłużysz czas trwania testów nie zostały osiągnięte żadne rezultaty. Należy tutaj zwrócić uwagę na to że Redis to magazyn danych w pamięci, który jest używany jako pamięć podręczna. Ma szybki czas reakcji i przetwarza miliony żądań w czasie rzeczywistym. Jednak jego wadą jest to że wymaga w tym celu ogromnej pamięci RAM. Jak pokazują zamieszczone wyniki przy dodawaniu małej liczby rekordów Redis osiągnął podobne czasy jak pozostałe bazy. Jednak przy dodaniu 1000 rekordów czas drastycznie wzrósł w porównaniu do pozostałych baz. Ze względu na brak możliwości dodania wszystkich rekordów w rozsądnym czasie, Redis nie został uwzględniony w dalszej części badań.



2. Operacja usuwania rekordów

Kolejne badanie obejmowało pomiar wydajności operacji usuwania rekordów z bazy MongoDB i PostgreSQL.

Usuwanie danych zostało przeprowadzone po wcześniejszym dodaniu wszystkich rekordów do bazy tak aby sprawdzić wydajność baz, kiedy są obciążone dużą ilością danych. Jak można zauważyć przy operacji usuwania małej liczby rekordów obie bazy mają zbliżone czasu. Operacja ta przebiega sprawnie. Natomiast przy usuwaniu 1000 rekordów widać, że PostgreSQL zajmuję to znacznie więcej czasu niż MongoDB dla którego czas trwania tej operacji nie uległ drastycznemu pogorszeniu.

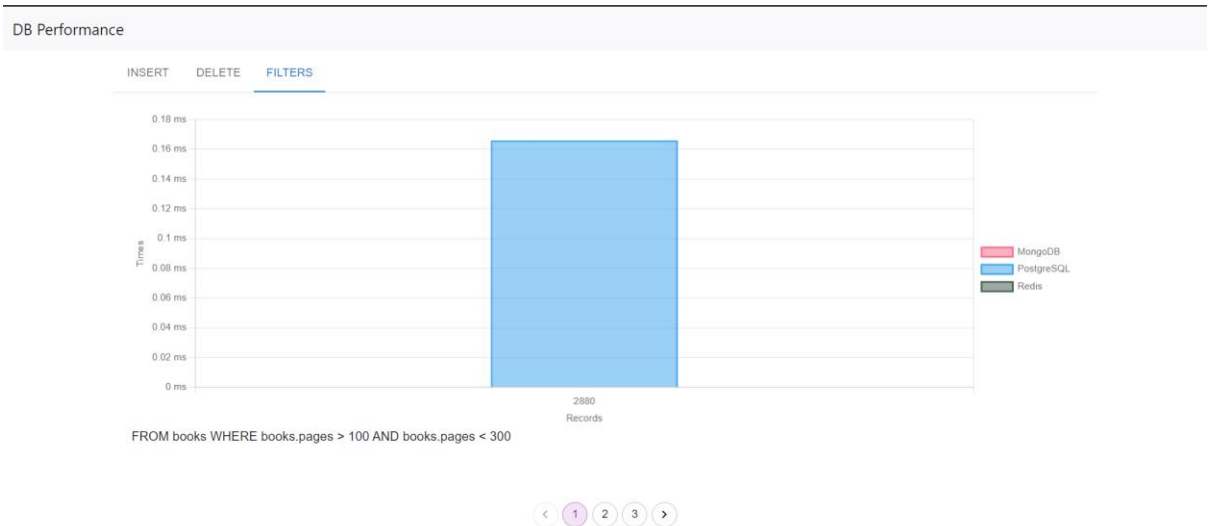


3. Operacja wybierania rekordów

To badanie polegało na pomiarze wydajności operacji wybierania rekordów z różnymi filtrami z baz MongoDB i PostgreSQL. W legendzie widocznej we frontendzie jest również widoczna baza Redis, ale nie jest ona wykorzystywana w tych testach ze względu na czas dodawania rekordów. Zostały przeprowadzone 3 testy wybierania rekordów z filtrami o różnym skomplikowaniu. Przed każdym pomiarem do baz danych są dodawane wszystkie rekordy.

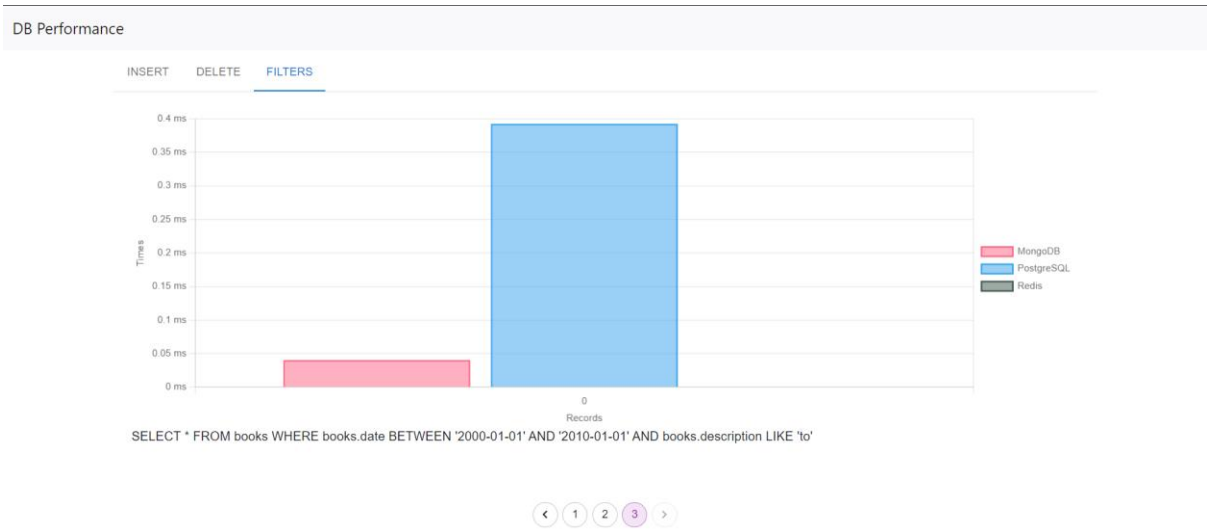
Pierwsze zapytanie jest dosyć proste i możemy zaobserwować, że jest ono wykonywane natychmiastowo w przypadku bazy MongoDB. W przypadku bazy PostgreSQL zapytanie jest wykonywane również dosyć szybko (około 0.16ms), ale jest to czas o wiele dłuższy niż w MongoDB.

Zapytanie



Drugie zapytanie jest bardziej skomplikowane od pierwszego i zajmuje więcej czasu obu bazom, ale jest wykonywane kilkukrotnie szybciej w przypadku bazy MongoDB.

Trzecie zapytanie pokazało jeszcze większą przewagę bazy MongoDB nad PostgreSQL.



Wszystkie zapytania są wykonywane znacznie szybciej w bazie MongoDB niż w PostgreSQL.

10. Wnioski

Celem projektu było pozyskanie znacznej ilości danych, przeprowadzenie badań wydajności bazy o modelu relacyjnym i dwóch baz o modelu nierelacyjnym przy użyciu pozyskanych danych oraz stworzenie interfejsu użytkownika w celu prezentacji uzyskanych wyników.

Badania wydajności zostały przeprowadzone dla operacji dodawania, usuwania i wybierania rekordów z baz danych. Udało się przeprowadzić wszystkie badania dla bazy MongoDB oraz PostgreSQL. Baza Redis ze względu na brak rezultatów dodawania dużej

ilość rekordów z rozsądnym czasie została z dalszych testów odrzucona. Wobec tego nie udało się przeprowadzić wszystkich badań wydajności dla bazy Redis.

Na podstawie uzyskanych rezultatów można zauważyć, że model relacyjny jest wydajny dla małej liczby rekordów natomiast przy znacznie większej liczbie oraz skomplikowanych relacjach model nierelacyjny w tym przypadku MongoDB jest wydajniejszy.