

Podsumowanie laboratorium

PORÓWNANIE BAZY DANYCH MARIA Z MONGO

Jakub Wąsik | Bazy danych NoSQL | 12.05.20

Link do kodu na githubie: <https://github.com/KubaWasik/NoSQLProject>

Lab 1 - MariaDb

W pierwszej części zadania należało przetestować bazę danych Maria, zgodnie z poleceniem testowałem wydajność dla:

- Tworzenia bazy danych Maria i zapełniania jej danymi – plik lab1_create (https://github.com/KubaWasik/NoSQLProject/blob/master/lab1_create.py)
- Migracji bazy danych (1 tabeli) z bazy Maria do bazy Mongo – plik lab1_migrate (https://github.com/KubaWasik/NoSQLProject/blob/master/lab1_migrate.py)
- Testowanie wydajności podanego na zajęciach algorytmu o złożoności wykładniczej dla dwóch tabel w bazie Maria – plik lab1_algorithm (https://github.com/KubaWasik/NoSQLProject/blob/master/lab1_algorithm.py)

Testy miały zostać zautomatyzowane oraz wykonane dla 10^N rekordów, ze względu czas ostatni krok testów ograniczyłem do N maksymalnie wynoszącego 4. Testy zostały uruchomione kilkukrotnie wyniki są średnimi wyników uzyskanych z testu lub ich przybliżeniami. Plik wykorzystany do automatyzacji testów – lab1.sh (<https://github.com/KubaWasik/NoSQLProject/blob/master/lab1.sh>)

N	Tworzenie bazy danych i tabeli	Migracja MariaDb -> MongoDB	Algorytm
1	0,034784 s	0,032962 s	0,597203 s
2	0,038214 s	0,039309 s	0,841378 s
3	0,121013 s	0,998739 s	9,667950 s
4	0,924258 s	0,669089 s	515,522300 s
5	8,601578 s	6,750162 s	-
6	85,481554 s	66,684128 s	-
7	906,915741 s	779,884519 s	-

Lab 2 - MongoDB

W drugiej części należało przetestować wydajność bazy danych Mongo, również przetestowałem ją według ustalonego schematu:

- Tworzenia bazy danych Mongo i zapełniania jej danymi – plik lab2_create (https://github.com/KubaWasik/NoSQLProject/blob/master/lab2_create.py)
- Migracji bazy danych (1 tabeli) z bazy Maria do drugiej bazy Mongo – plik lab2_migrate (https://github.com/KubaWasik/NoSQLProject/blob/master/lab2_migrate.py)
- Testowanie wydajności podanego na zajęciach algorytmu o złożoności wykładniczej dla dwóch tabel w bazie Mongo – plik lab2_algorithm (https://github.com/KubaWasik/NoSQLProject/blob/master/lab2_algorithm.py)

Tak samo jak w Lab 1 testy zostały zautomatyzowane a wyniki dla algorytmu dla N większego niż 4 pominięte. Plik wykorzystany do automatyzacji testów – lab2.sh (<https://github.com/KubaWasik/NoSQLProject/blob/master/lab2.sh>)

N	Tworzenie bazy danych i tabeli	Migracja MongoDB -> MongoDB	Algorytm
1	0,028434 s	0,035112 s	0,074317 s
2	0,038766 s	0,040051 s	0,644696 s
3	0,109842 s	0,079403 s	9,692895 s
4	0,837904 s	0,473344 s	751,453522 s
5	9,870632 s	4,605177 s	-
6	86,385705 s	48,953757 s	-
7	899,394826 s	579,298601 s	-

Wnioski

Na początku należy zaznaczyć, że wyniki nie są dokładne, wynika to z faktu użycia wirtualnej maszyny, która dynamicznie udziela zasobów głównego komputera dla systemów wirtualnych, dodatkowo występowały pewne przekłamania mierzonego czasu, polecenie *time* systemu Linux pokazywało diametralnie różne wyniki od czasomierza Python-a. Sytuacja testowa nie może jednoznacznie określić która baza danych jest szybsza.

Z wyżej ukazanych wyników można zauważyć, że migracja w obu przypadkach ma liniową złożoność lecz baza Mongo jest w tym przypadku zauważalnie szybsza, tworzenie bazy danych w oby przypadkach zależna była od losowania ciągów znaków, które dodatkowo były sprawdzane pod kątem duplikatów, co przy dużej ilości ciągów mogło sprawiać że wyniki były bardzo rozbieżne lecz średnia tych wyników jest bliska siebie w obu przypadkach. Wymyślony na zajęciach algorytm nie jest dobrym wyznacznikiem wydajności lecz w sytuacji dla N równego 4 czyli dla 10000 rekordów, program dla bazy Mongo był widocznie wolniejszy.