

ZTBD



Sprawozdanie

Projekt

MySQL i Elasticsearch

Autorzy: Patryk Świerkowski, Jakub Piątek

Wprowadzenie

Zadaniem bazy danych jest przechowywanie, organizacja i wyszukiwanie odpowiednio uporządkowanych danych.

Wyróżniamy bazy danych relacyjne(RDBMS) i nie relacyjne(noSQL).

Relacyjna baza danych jest to zbiór pozycji danych które są zorganizowane jako zbiór tabel opisanych pewnymi relacjami. Każda tabela składa się z rekordów. Pojedynczy rekord nazywamy krotką.

Cechy relacyjnej bazy danych:

- używa języka SQL
- składa się z tabel
- wiersze(rekordy) w danej tabeli mają identyczną strukturę(spójność danych)
- relacyjność, umożliwia łączenie tabel za pomocą odpowiednio zdefiniowanych kluczy
- skalowanie wertykalne,
- pozwala na długie i kompleksowe zapytania

SQL(Structured Query Language) jest to ustrukturyzowany język zapytań w bazach danych typu sql(relacyjnych).

Przykład zapytania SQL

```
SELECT id, email, name FROM users
```

To zapytanie SQL pobiera id , email i nazwę wszystkich rekordów z tabeli użytkowników.

RDBMS(Relational database Management System) jest to system zarządzania relacyjną bazą danych.

Najpopularniejszymi RDBMS są:

- MySQL
- PostgreSQL
- Oracle Database

MySQL - jest open sourceowym systemem do zarządzania relacyjną bazą danych. Z MySQL korzystają najpopularniejsze witryny i aplikacje internetowe na świecie, w tym Uber, Facebook, Youtube.

Jedną z zalet MySQL jest ochrona danych ponieważ posiada wiele mechanizmów ochrony danych takich jak uwierzytelnienie i autoryzacje użytkownika i wsparcie dla protokołów ssh czy ssl.

noSQL(not Structured Query Language) jest to baza danych typu noSQL która nie używa zapytań SQL. Jest to nie relacyjna baza danych.

Cechy nie relacyjnej bazy danych?

- składa się z Kolekcji(Collections)

- każdy dokument w danej kolekcji jest od siebie niezależny, dokument wygląda jak obiekt JSON.

- skalowalność horyzontalna i wertykalna, możemy dodawać kolejne serwery i ulepszającą dany serwer.

- działa szybko nawet z big data

Najpopularniejszymi bazami noSQL są:

- MongoDb

- ElasticSearch

Elastic Search

Elastic Search jest bazą NoSQL, która została zbudowana z biblioteki Apache Lucene. Jest to rozwiązanie Open Source. Ze względu na strukturę encji, którą charakteryzuje składnia JSON jest dokumentową bazą danych.

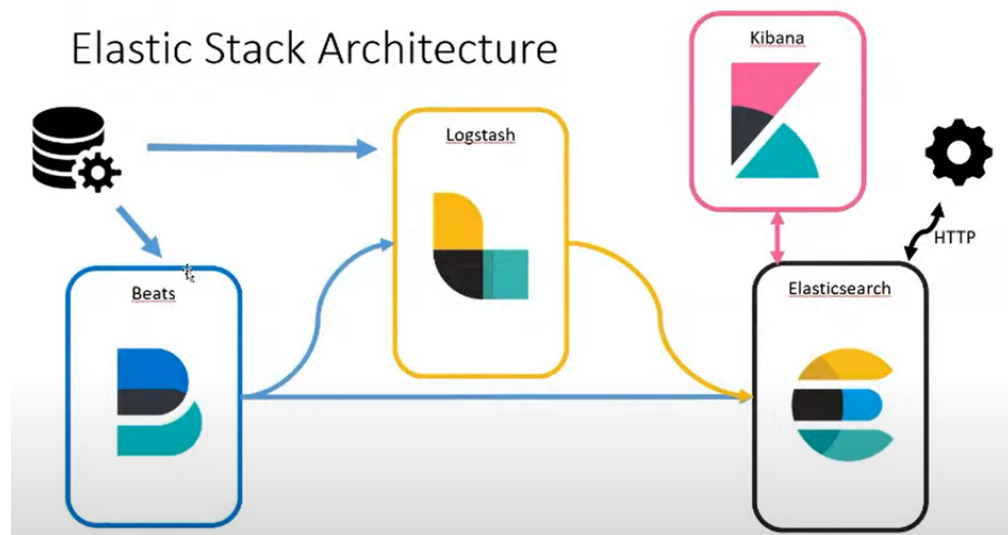
Jest jednym z elementów oprogramowania Elastic Stack.

Elastic Stack

Elastic Stack jest to oprogramowanie do analizy logów, analizy danych w czasie rzeczywistych takich jak wyniki sprzedażowe , szybkiego wyszukiwania sekwencji znaków w wyszukiwarce.

Elastic Stack składa się z takich elementów jak Beats, Logstash, ElasticSearch, Kibana.

Architektura:



Alternatywą jest zastąpienie Beats i Logstash odpowiednią aplikacją np w PHP która umożliwi komunikację innej bazy danych np MySQL z Elastic search. Takie rozwiązanie umożliwiłoby sprawdzenie wydajności zapytań SQL na tych bazach danych, jeżeli posiadałyby taki sam zestaw danych.

Kibana

Kibana jest to narzędzie umożliwiające odpowiednie zarządzanie i wizualizację danych poprzez komunikację przez protokół REST API z bazą danych elastic search.

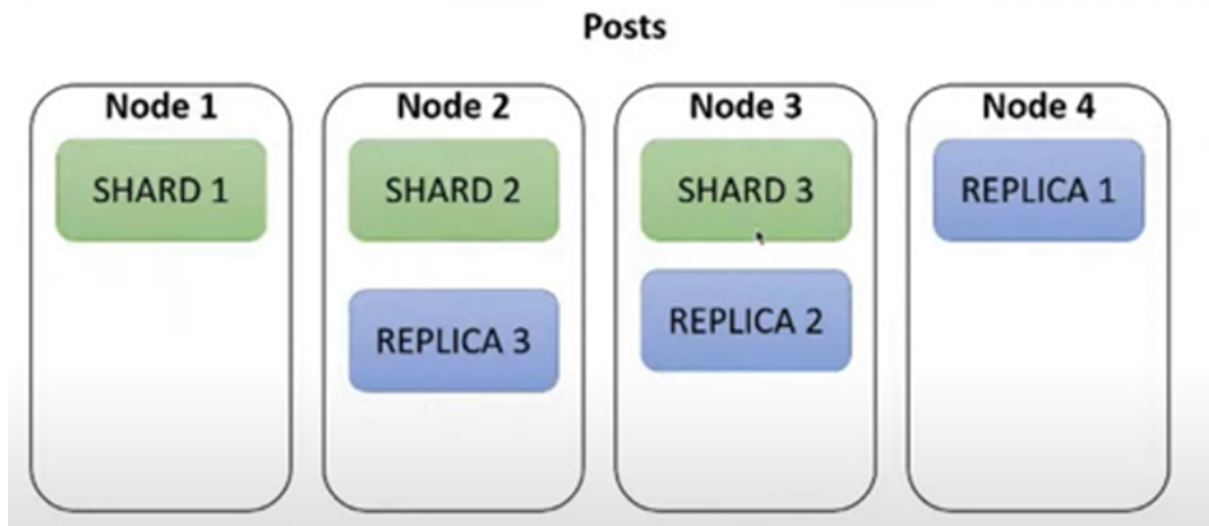
Porównanie relacyjnej bazy danych z elastic search:

RBD	Elasticsearch
Database	Index
Table	Type / Index
Row	Document
Column	Field
Schema	Mapping

Żeby stworzyć Index(Klaster) w Elastic Search musimy stworzyć instancje(nody) w których zostaną równomiernie rozmieszczone shardy. Elastic Search jest rozwiązaniem skalowalnym więc jest możliwość zwiększania ilości instancji.

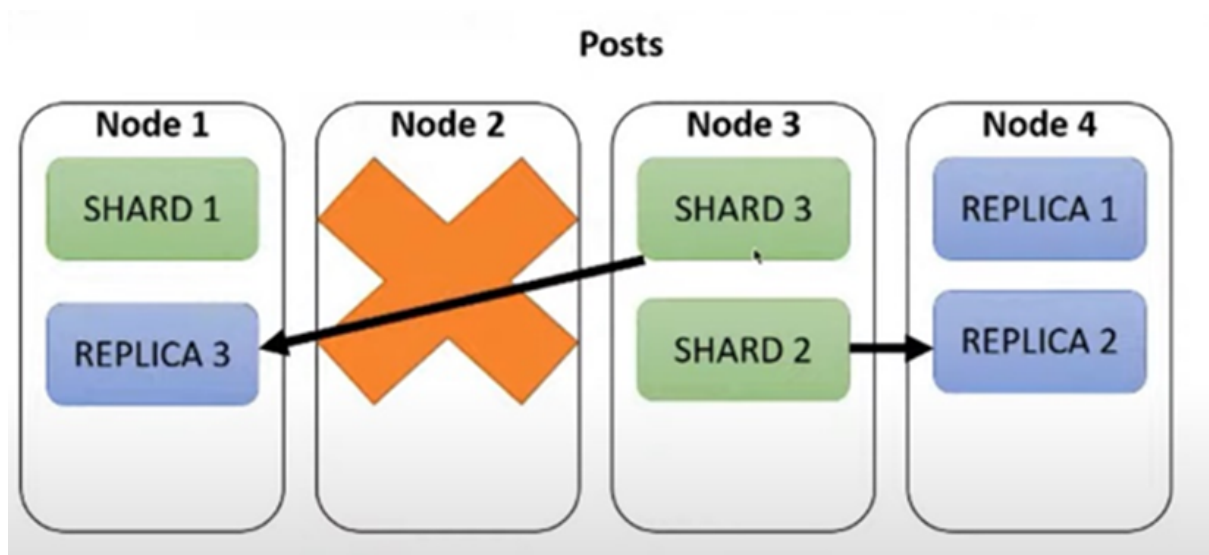
Każdy Shard może się dynamicznie replikować n-razy dlatego ilość instancji może zostać zwiększona.

Przykład:



Zaletą takiego rozwiązania, zakładając że instancje(nody) są rozmieszczone na różnych serwerach jest ochrona danych w wyniku wystąpienia awarii na jednym z nodów.

Przykład:



Shardy mogą posiadać jedną z poniższych ról:

- Master
- Data
- Coordinator

Rola Master - oznacza że dany node ma zarządzać metadanymi. W klastrze powinna być nieparzysta ilość ról Master przypisanych do poszczególnych shardów.

Rola Data – oznacza że dany node ma przechowywać dane

Rola Coordinator- oznacza że dany node jest odpowiedzialnym za przyjmowanie danych i obsługę zapytań.

Celem użycia elastic search jest przechowywanie dużych zbiorów danych i ich szybkie wyszukiwanie. Dlatego posiada silnik analityczny, który umożliwia agregację podobnych zakresów danych spośród ogromnej ilości rekordów.

PHP

Jest to skryptowy język programowania służący do tworzenia stron internetowych. PHP jest rozprowadzany na otwartej licencji i każdy może pobrać za darmo jego kopię, zainstalować i używać bez żadnych ograniczeń zarówno do celów prywatnych jak i komercyjnych. Język jest prosty w nauce i umożliwia tworzenie profesjonalnych dynamicznych stron internetowych.

PHP jest językiem server-side, tj. pracuje po stronie serwera WWW. Przeciwnieństwem są języki client-side pracujące po stronie przeglądarki użytkownika (np. JavaScript). Aby wykorzystywać go na własnej stronie, musisz upewnić się, że twój serwer obsługuje PHP. Zanim przejdziemy dalej, należy zrozumieć zasadę, na jakiej PHP generuje dynamiczne strony.

Kiedy wpisujemy adres w przeglądarce internetowej, żądanie wyświetlenia strony kierowane jest do serwera HTTP zwanego także serwerem WWW. Jeśli stwierdzi na podstawie rozszerzenia pliku, że dany dokument zawiera kod PHP, serwer kieruje do jego interpretera żądanie przetworzenia podanego pliku. Interpreter wyszukuje w jego treści tzw. wstawki

PHP wplecione w statyczny kod HTML i zastępuje je wynikiem ich wykonywania. Utworzony kod HTML jest zwracany serwerowi, a ten wysyła go z powrotem do internauty. PHP używany jest do dynamicznego (zależnego od różnych warunków) generowania kodu HTML - zawartości strony. Do przeglądarki dociera kod HTML a nie PHP. Jeśli mamy plik PHP o następującej treści:

```
<html>
<body>
<?php
    echo 'Podaj hasło';
?>
</body>
</html>
```

To internauta zobaczy jedynie dokument o takiej treści:

```
<html>
<body>
Podaj hasło
</body>
</html>
```

Cały PHP zniknie, a na jego miejscu pojawi się utworzony przez niego kod HTML.

Dzięki pracy po stronie serwera, PHP idealnie nadaje się do tworzenia złożonych aplikacji zarządzających dużymi ilościami danych: forami dyskusyjnymi, systemami zarządzania treścią, sklepami internetowymi. Generują one odpowiedni kod HTML dla przeglądarek internautów, a w momencie, kiedy oni go przeglądają, PHP już zakończył nad nim swą

pracę. Jest to bardzo istotne, ponieważ wszelkie dalsze reakcje na poczynania użytkownika należy albo pozostawić przeglądarce, albo obsłużyć je za pomocą języka JavaScript.

MySQL Workbench

Jest to wizualne narzędzie do projektowania baz danych, które integruje programowanie, administrowanie, projektowanie i tworzenie baz danych SQL oraz zarządzanie nimi w jednym zintegrowanym środowisku programistycznym dla systemu baz danych MySQL

Należy o tym również wspomnieć dostępna jest wersja płatna, pozwoli nam to na korzystanie z NoSQL lub jeśli będziemy potrzebować dodatkowych funkcji, takich jak integracja z pakietem Office lub pomoc.

Doctrine ORM3

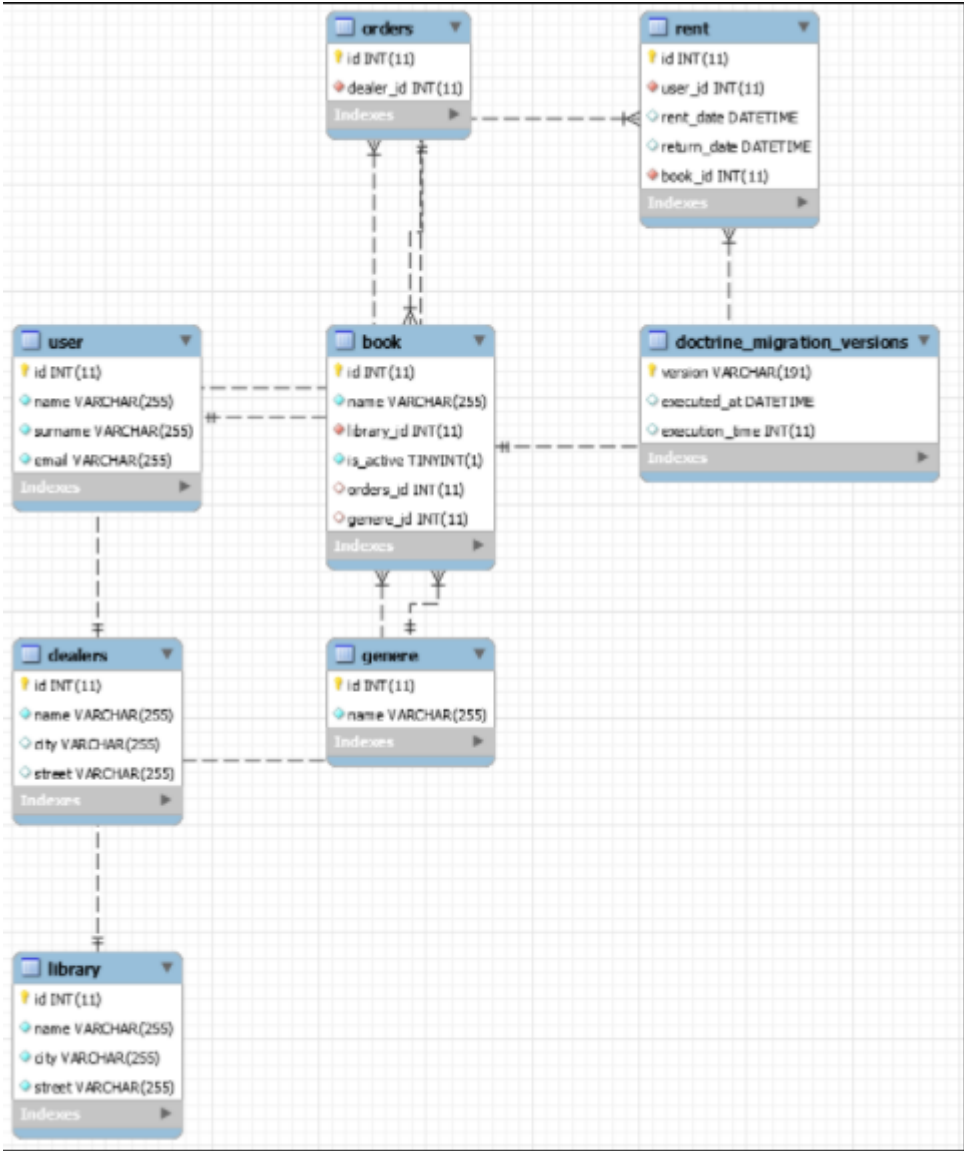
Doctrine ORM jest maperem obiektowo-relacyjnym (ORM) dla PHP 7.1+, który zapewnia przezroczystą trwałość obiektów PHP. W sercu wykorzystuje wzorzec Data Mapper, którego celem jest całkowite oddzielenie logiki domeny/biznesu od trwałości w systemie zarządzania relacyjnymi bazami danych.

Zaletą Doctrine dla programisty jest możliwość skupienia się na logice biznesowej zorientowanej obiektowo i martwienia się o trwałość tylko jako drugorzędny problem. Nie oznacza to, że Doctrine 2 bagatelizuje trwałość, jednak wierzymy, że istnieją znaczne korzyści dla programowania obiektowego, jeśli trwałość i byty są utrzymywane w separacji.

Symfony

Symfony to open-sourcowy framework oparty na PHP. Jest on bardzo popularny wśród programistów, gdyż znacznie redukuje czas potrzebny na tworzenie oprogramowania. ... Dzięki efektywnemu tworzeniu kodu programiści mogą tworzyć jakościowe aplikacje w stosunkowo szybkim czasie i przystępnym kosztem.

Schemat bazy danych wygenerowany MySQL Workbench



Indexy stworzone w Elasticsearch:

<input type="checkbox"/> Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
<input type="checkbox"/> app	● yellow	open	1	1	0	226b	
<input type="checkbox"/> book	● yellow	open	1	1	73800	5.7mb	
<input type="checkbox"/> dealers	● yellow	open	1	1	10000	540.2kb	
<input type="checkbox"/> rent	● yellow	open	1	1	101	975.8kb	
<input type="checkbox"/> library	● yellow	open	1	1	10	41.6mb	
<input type="checkbox"/> user	● yellow	open	1	1	11850	2.1mb	

Zalety i wady wybranych baz

	Elastic Search	MySQL
Zalety	<ul style="list-style-type: none"> - wysoka skalowalność - dobrze radzi sobie z BigData - możliwość rozproszenia bazy na wielu serwerach 	<ul style="list-style-type: none"> -skalowalny i wydajny -szybki - łatwy w obsłudze - bezpieczeństwo -open Source DBMS - może być zintegrowany z różnymi językami programowania -przenośne oprogramowanie - uniwersalność zapytań i operacji SQL
Wady	<ul style="list-style-type: none"> - Elastic nie jest typową bazą noSQL. Dobrze sprawdza się jako wyszukiwarka, ale ma problemy np z zakładaniem encji, a indexy potrafią odświeżać się dłuższy czas co jest problematyczne przy usuwaniu danych 	<ul style="list-style-type: none"> -brak implementacji pełnego standardu SQL -słaba pomoc techniczna

Obszary zastosowania wybranych baz

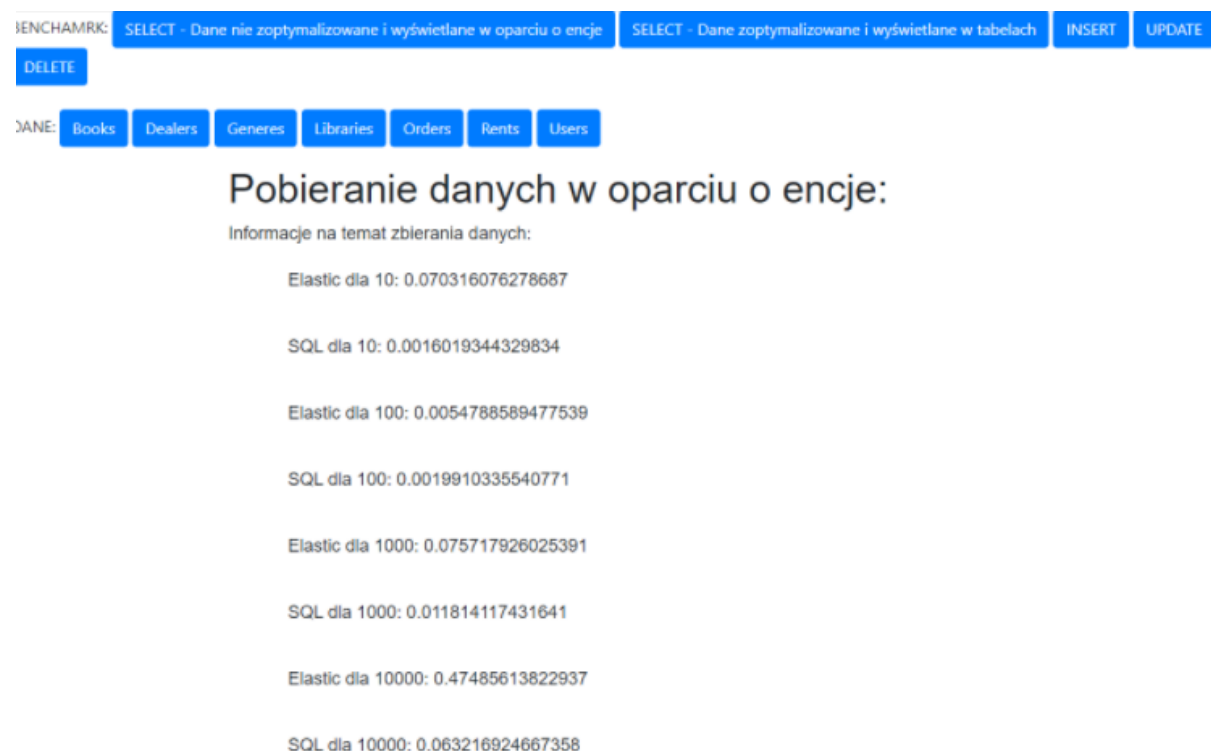
Elastic Search	MySQL
<ul style="list-style-type: none"> - pełnotekstowe wyszukiwanie tekstu - blogi i systemy zarządzania treścią - zbieranie i analiza logów - zbieranie i analiza danych dostępnych publicznie - zbieranie i analiza danych z pomiarów - wizualizacja danych - blogi i systemy zarządzania treścią 	<ul style="list-style-type: none"> - ecommerce - strony internetowe i aplikacje - używany w systemach produkcyjnych o dużym obciążeniu

Ocena wydajności baz:

Crud

Select - dla danych nie zoptymalizowanych i wyświetlanych w oparciu o encje

W przypadku bazy ElasticSearch to w przypadku naszej konfiguracji nie działa ona stabilnie, gdy próbujemy budować encje tylko poprzez elasticsearch, a więc wymagany do ich założenia jest również SQL



Select - dla danych zoptymalizowanych i wyświetlanych w tabelach

DELETE

Pobieranie danych w oparciu o wyszukiwanie czystych rekordów:

Informacje na temat zbierania danych:

Elastic dla 10:	0.028956890106201
SQL dla 10:	0.020983934402466
Elastic dla 100:	0.0027401447296143
SQL dla 100:	0.00060081481933594
Elastic dla 1000:	0.010931015014648
SQL dla 1000:	0.0025758743286133
Elastic dla 10000:	0.10856986045837
SQL dla 10000:	0.01778507232666

Insert

DELETE

Insert:

Informacje na temat wykonywania danych:

Elastic dla 10:	0.028915166854858
SQL dla 10:	0.16684699058533
Elastic dla 100:	0.011759996414185
SQL dla 100:	0.036687850952148
Elastic dla 1000:	0.048565864562988
SQL dla 1000:	0.19353890419006
Elastic dla 10000:	0.42742300033569
SQL dla 10000:	1.8519749641418

Delete

Usuwanie rekordów:

Informacje na temat usuwania danych:

Elastic dla 10: 0.11791515350342

SQL dla 10: 0.004662036895752

Elastic dla 100: 5.0067901611328E-6

SQL dla 100: 0.50462508201599

Elastic dla 1000: 20.605561971664

SQL dla 1000: 3.2091319561005

Elastic dla 10000: 175.47705411911

SQL dla 10000: 4.9724440574646

Update

Update:

Informacje na temat wykonywania danych:

Elastic dla 10: 0.15187406539917

SQL dla 10: 0.040837049484253

Elastic dla 100: 1.1745347976685

SQL dla 100: 0.023890018463135

Elastic dla 1000: 12.483926057816

SQL dla 1000: 0.20124292373657

Elastic dla 10000: 127.75062704086

SQL dla 10000: 3.1133000850677

Cel projektu

Celem projektu było wskazanie różnic pomiędzy bazą relacyjną MySQL a Elasticsearch przy użyciu aplikacji webowych.

W tym celu została zaimplementowana aplikacja testująca wydajność tych baz.

Realizacja projektu

Zastosowane technologie

- Symfony 5
- PHP 8
- Elasticsearch 7
- Kibana
- DBEaver
- MySQL Workbench
- MySQL
- Doctrine ORM 3
- FOS Elastica

Środowisko:

- Kibana
- PHPStorm

Funkcjonalność:

aplikacja testująca zawiera mechanizm transakcji, system uwierzytelnienia użytkowników, mechanizm bezpieczeństwa , system oceny wydajności.

Instrukcja uruchomienia

Instalacja PHP - wersja 8.0 lub wyższa

Instalacja MySQL - w projekcie został wykorzystany XAMPP

Instalacja rozsz

Instalacja ElasticSearch

Instalacja Kibana

Instalacja Composer

Instalacja Node wraz z rozszerzeniem NPM lub Yarn

Uruchomienie ElasticSearch i Kibana

Użycie instrukcji w terminalu w głównym katalogu **composer install**

Konfiguracja pliku .env - dodanie linii DATABASE_URL i ELASTICSEARCH_URL z konfiguracją. Konfiguracje zastosowane w projekcie:

```
DATABASE_URL="mysql://root:@127.0.0.1:3306/db_name?serverVersion=10.4.22-MariaDB"
ELASTICSEARCH_URL=http://localhost:9200/
```

W celu pełnego działania systemu wymagane jest ustawienie przynajmniej 6GB RAM.

Używane do wygenerowania pliku fixtures.

Następnie uruchamiamy server przez komendę: **symfony server:start**

W celu uruchomienia serwera w tle można dołączyć flagę -d.

Następnie wykonujemy migracje poprzez użycie polecenia: **php bin/console do:mi:migrate**

Po poprawnym wykonaniu migracji wykonujemy komendę **php bin/console**

fos:elastica:populate

Zasili ona bazę danych w elasticsearch. Komenda wykonuje się bardzo długo i nie jest zoptymalizowana pod względem kolejkowania, a więc dane powinny ładować się około 2 godzin.

W przypadku zmian frontowych uruchamiamy **npm run:watch** w innym wypadku wysarczy **npm:run:serve**

W przypadku ustawień domyślnych powinniśmy mieć dostępne:

<https://localhost:9200> - elasticsearch

<https://localhost:8000> - aplikacja

<https://localhost:5601> - kibana

Wnioski

Dane dotyczą tego w jaki sposób informacje zbierane są w aplikacji, a nie strikte samodzielnych zapytań co ma na celu pokazanie realnej użyteczności.

DATABASE	ROWS	UPDATE	SELECT	SELECT (encja)	INSERT	DELETE
MySQL	10	0,040	0,020	0,001	0,166	0,004
	100	0,023	0,000	0,001	0,036	0,504
	1000	0,201	0,002	0,011	0,193	3,209
	10000	3,113	0,017	0,063	1,851	4,972
ElasticSearch	10	0,151	0,028	0,070	0,028	0,117
	100	1,174	0,002	0,005	0,011	5,006
	1000	12,483	0,010	0,075	0,048	20,605
	10000	127,750	0,108	0,474	0,427	175,477

Dla oceny wartości strikte na zapytaniach wykorzystany został profiler wbudowany we framework symfony.

Przykładowy widok z profillera:

localhost:8000/_profiler/86600f?panel=db

Symfony Profiler

search on symfony.com Search

http://localhost:8000/index

Method: GET HTTP Status: 200 IP: ::1 Profiled on: Sun, 30 Jan 2022 17:38:59 +0100 Token: 86600f

Last 10 Latest Search

Request / Response

Performance

Validator

Forms

Exception

Logs 10

Events

Routing

Cache

Translation

Security

Twig

HTTP Client

Doctrine

Debug

E-mails

Notifications

Elastica

Query Metrics

8	5	79.65 ms	0
Database Queries	Different statements	Query time	Invalid entities

Queries

Group similar statements

#	Time	Info
1	5.42 ms	<pre>SELECT b0_.id AS id_0, b0_.name AS name_1, b0_.is_active AS is_active_2, b0_.library_id AS library_id_3, b0_.orders_id AS orders_id_4, b0_.genre_id AS genre_id_5 FROM book b0_ WHERE b0_.id IN (?)</pre> <p>Parameters:</p> <pre>[]</pre> <p>View formatted query View runnable query Explain query</p>
2	0.53 ms	<pre>SELECT b0_.id AS id_0, b0_.name AS name_1, b0_.is_active AS is_active_2, b0_.library_id AS library_id_3, b0_.orders_id AS orders_id_4, b0_.genre_id AS genre_id_5 FROM book b0_ LIMIT 10</pre> <p>Parameters:</p> <pre>[]</pre> <p>View formatted query View runnable query Explain query</p>
3	0.61 ms	<pre>SELECT b0_.id AS id_0, b0_.name AS name_1, b0_.is_active AS is_active_2, b0_.library_id AS library_id_3, b0_.orders_id AS orders_id_4, b0_.genre_id AS genre_id_5 FROM book b0_ WHERE b0_.id IN (?)</pre> <p>Parameters:</p> <pre>[]</pre> <p>View formatted query View runnable query Explain query</p>
4	0.54 ms	<pre>SELECT b0_.id AS id_0, b0_.name AS name_1, b0_.is_active AS is_active_2, b0_.library_id AS library_id_3, b0_.orders_id AS orders_id_4, b0_.genre_id AS genre_id_5 FROM book b0_ LIMIT 100</pre> <p>Parameters:</p> <pre>[]</pre> <p>View formatted query View runnable query Explain query</p>

Projekt został wysłany bez folderów vendor i node_modules, ponieważ można je pobrać z menadżerów paczek.