

Міністерство освіти і науки України  
Національний технічний університет України „КПІ”  
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки  
інформації та управління

ЗВІТ

з лабораторної роботи № 5  
дисципліни  
“ТЕХНОЛОГІЇ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ В УМОВАХ  
ВЕЛИКИХ ДАНИХ”  
на тему:

„Контейнеризація рішення”

Виконали:  
студенти групи ІТ-01мн  
Панасюк Станіслав  
Лесогорський Кирило

Перевірив:  
доц. Жереб К. А.

## Зміст

1. Постановка задачі .....	3
2. Обрані інструменти.....	3
3. Опис контейнеризації .....	4
4. Опис додаткового функціоналу .....	5
5. Отримані результати.....	6
6. Висновки .....	6

## 1. Постановка задачі

Для обраної задачі необхідно реалізувати послідовну (однопоточну) реалізацію, а також мультипоточну реалізацію зі спільною пам'яттю. У якості задачі було обрано побудову системи пошуку схожих зображень. У ядрі системи лежатиме використання D-hash для знаходження хешу зображення. D-hash дозволяє точно та швидко шукати схожі зображення. Він стійкий до скейлінгу зображення, але погано справляються з обрізаними та повернутими під кутом зображеннями. Тому цю техніку аугментовано за допомогою наступного прийому: при завантаженні зображення воно буде аугментовано за допомогою декількох фільтрів, при цьому для кожного фільтра буде згенеровано хеш і збережено у базу даних. При пошуку зображення буде використовуватись оператор XOR для знаходження зображень зі схожими хешами. Метою цієї лабораторної роботи є контейнеризація додатку та доробка функціоналу пошуку вже існуючих зображень.

## 2. Обрані інструменти

Окрім вже використаних технологій, ми скористаємось Docker та Docker Compose. З такою конфігурацією додаток буде легко розгорнути у клауд провайдері, наприклад AWS з сервісом Elastic Beanstalk. Також буде використано специфіку JPA – DTO Projection для мапінгу результатів нативного запиту у базу даних у інтерфейс для подальшої обробки у додатку.

### 3. Опис контейнеризації

Для контейнеризації було створено докер зображення для java сервісу та docker compose файл для оркестрації БД, брокеру черги та двох сервісів. Докер зображення базується на зображенні з Gradle версії 7.3, яке в свою чергу базується на OpenJDK 11 та Alpine linux. Файл можна переглянути у [репозиторії](#), але для наочності додамо скріншот:

```
1 FROM gradle:7.3-jdk11-alpine
2 COPY . /home/gradle/src
3 WORKDIR /home/gradle/src
4 RUN mkdir /imrec-images
5 EXPOSE 8080
6
7 ENTRYPOINT ["gradle", "bootRun"]
```

Рис. 3.1 – Dockerfile

Одразу видно що цей файл неоптимальний і його можна покращити. Для цього слід додати `.dockerignore` для запобігання зайвого копіювання та використати multi-stage build, де на першому етапі ми компілюємо сервіс у jar файл, а на другому етапі використовуємо отриманий файл на більш легковісному базовому зображенні без інструментів необхідних для компіляції сервісу.

Далі слід розглянути [docker-compose](#). У ньому ми визначаємо 4 сервіси: інжестор, воркер, брокер черги та СУБД.

Для СУБД ми використовуємо Postgres 14. До нього монтується вольюм для персистентного збереження даних між рестартами контейнера. Також ми використовуємо змінну оточення щоб вказати адміністраторський пароль та директиву ports для того, щоб отримати доступ до СУБД ззовні docker.

Для брокера черги ми використовуємо RabbitMQ. До нього також монтується вольюм для персистентного збереження даних. Для зовнішнього доступу відкривається порт 15672, на якому розгортається менеджер брокеру для налаштувань.

Далі розгортається два сервіси. Вони використовують однакове базове зображення докер та до них монтується спільний вольюм для завантаження та обробки зображень. Режим запуску контролюється за допомогою змінної оточення `SPRING_ACTIVE_PROFILE`, що дозволяє вказати у якому профілі слід запускати додаток.

```

1  version: "3"
2
3  services:
4    injector:
5      build: .
6      ports:
7        - 8080:8080
8      volumes:
9        - images:/imrec-images
10     environment:
11       - SPRING_PROFILES_ACTIVE=injector
12     depends_on:
13       - postgres
14       - queue
15     queue:
16       image: rabbitmq:3.9-management
17       ports:
18         - 15672:15672
19       volumes:
20         - queuev:/var/lib/rabbitmq
21     worker:
22       build: .
23       volumes:
24         - images:/imrec-images
25       environment:
26         - SPRING_PROFILES_ACTIVE=worker
27       depends_on:
28         - postgres
29         - queue
30     postgres:
31       image: postgres:14
32       restart: always
33       ports:
34         - 5432:5432
35       environment:
36         POSTGRES_PASSWORD: secret

```

Рис. 3.2 – docker-compose

## 4. Опис додаткового функціоналу

У ході виконання роботи було реалізовано додатковий функціонал пошуку зображень за хешем. Для цього було створено додатковий [запит у репозиторії](#), відповідну [проекцію](#) даних, [сервіс](#) та метод [контроллера](#).

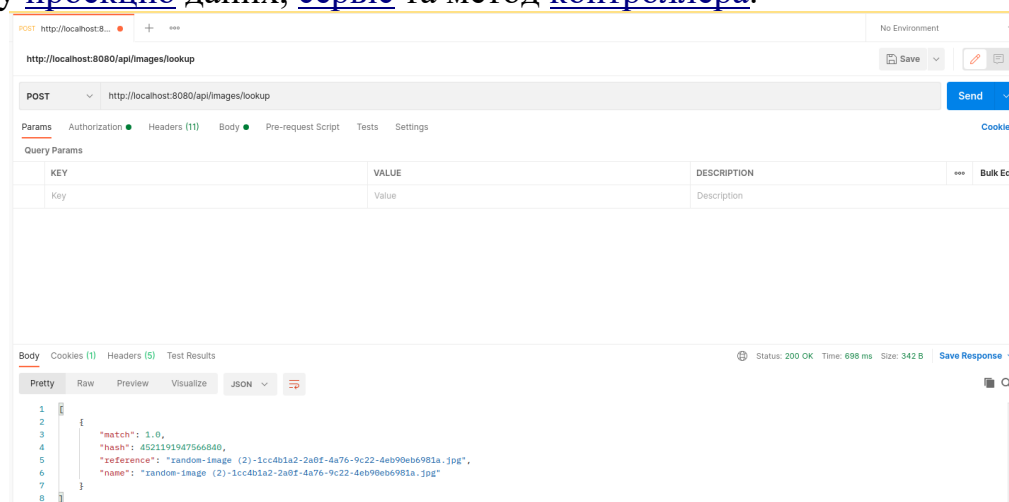


Рис. 4.1 – Пошук схожих зображень

## 5. Отримані результати

В результаті виконання лабораторної роботи додаток було контейнеризовано та додано новий функціонал.

Terminal:	Local	Local (2)	Local (3)	+
injector_1	2021-12-21 22:36:20.868	INFO 202	...	[main] org.hibernate.Version : HH0000412: Hibernate ORM core version 5.6.1.Final
worker_1	2021-12-21 22:36:20.992	INFO 201	...	[main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
worker_1	2021-12-21 22:36:21.010	INFO 201	...	[main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
worker_1	2021-12-21 22:36:21.010	INFO 201	...	[main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.55]
injector_1	2021-12-21 22:36:21.066	INFO 202	...	[main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations (5.1.2.Final)
worker_1	2021-12-21 22:36:21.163	INFO 201	...	[main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
worker_1	2021-12-21 22:36:21.164	INFO 201	...	[main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2660 ms
injector_1	2021-12-21 22:36:21.210	INFO 202	...	[main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
injector_1	2021-12-21 22:36:21.489	INFO 202	...	[main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
injector_1	2021-12-21 22:36:21.514	INFO 202	...	[main] org.hibernate.dialect.Dialect : HH0000400: Using dialect: org.hibernate.dialect.PostgreSQL100Dialect
worker_1	2021-12-21 22:36:21.675	INFO 201	...	[main] o.hibernate.jpa.internal.util.LogHelper : HH0000204: Processing PersistenceUnitInfo [name: default]
worker_1	2021-12-21 22:36:21.744	INFO 201	...	[main] org.hibernate.Version : HH0000412: Hibernate ORM core version 5.6.1.Final
worker_1	2021-12-21 22:36:21.895	INFO 201	...	[main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations (5.1.2.Final)
worker_1	2021-12-21 22:36:21.983	INFO 201	...	[main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
worker_1	2021-12-21 22:36:22.163	INFO 201	...	[main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
worker_1	2021-12-21 22:36:22.183	INFO 201	...	[main] org.hibernate.dialect.Dialect : HH0000400: Using dialect: org.hibernate.dialect.PostgreSQL100Dialect
injector_1	2021-12-21 22:36:22.218	INFO 202	...	[main] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
injector_1	2021-12-21 22:36:22.225	INFO 202	...	[main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
worker_1	2021-12-21 22:36:22.879	INFO 201	...	[main] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
injector_1	2021-12-21 22:36:22.878	WARN 202	...	[main] JpaBaseConfigurationJpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring
	.jpa.open-in-view to disable this warning			
worker_1	2021-12-21 22:36:22.889	INFO 201	...	[main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
injector_1	2021-12-21 22:36:23.460	INFO 202	...	[main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 13 endpoint(s) beneath base path '/actuator'
queue_1	2021-12-21 22:36:23.547852+00:00	[info]	<0.640.0>	accepting AMQP connection <0.640.0> (172.21.0.5:36800 -> 172.21.0.2:5672)
queue_1	2021-12-21 22:36:23.572878+00:00	[info]	<0.640.0>	Connection <0.640.0> (172.21.0.5:36800 -> 172.21.0.2:5672) has a client-provided name: rabbitConnectionFactory#719bb60d:0
queue_1	2021-12-21 22:36:23.583584+00:00	[info]	<0.640.0>	Connection <0.640.0> (172.21.0.5:36800 -> 172.21.0.2:5672) - rabbitConnectionFactory#719bb60d:0: user 'guest' authenticated and granted access to vhost '/'
injector_1	2021-12-21 22:36:23.533	INFO 202	...	[main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
injector_1	2021-12-21 22:36:23.536	INFO 202	...	[main] o.s.a.r.c.CachingConnectionFactory : Attempting to connect to: [queue:5672]
injector_1	2021-12-21 22:36:23.588	INFO 202	...	[main] o.s.a.r.c.CachingConnectionFactory : Created new connection: rabbitConnectionFactory#719bb60d:0/SimpleConnection@7e4dddee [delegate=amqp://guest@172.21.0.2:5672/, localPort= 36800
	]			
worker_1	2021-12-21 22:36:23.599	WARN 201	...	[main] JpaBaseConfigurationJpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring
	.jpa.open-in-view to disable this warning			
injector_1	2021-12-21 22:36:23.671	INFO 202	...	[main] lab.imrec.injector.InjectorApplication : Started InjectorApplication in 6.976 seconds (JVM running for 7.989)
worker_1	2021-12-21 22:36:24.166	INFO 201	...	[main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 13 endpoint(s) beneath base path '/actuator'
queue_1	2021-12-21 22:36:24.242751+00:00	[info]	<0.653.0>	accepting AMQP connection <0.653.0> (172.21.0.4:60394 -> 172.21.0.2:5672)
queue_1	2021-12-21 22:36:24.252269+00:00	[info]	<0.653.0>	Connection <0.653.0> (172.21.0.4:60394 -> 172.21.0.2:5672) has a client-provided name: rabbitConnectionFactory#788d9139:0
queue_1	2021-12-21 22:36:24.256647+00:00	[info]	<0.653.0>	Connection <0.653.0> (172.21.0.4:60394 -> 172.21.0.2:5672) - rabbitConnectionFactory#788d9139:0: user 'guest' authenticated and granted access to vhost '/'
worker_1	2021-12-21 22:36:24.229	INFO 201	...	[main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
worker_1	2021-12-21 22:36:24.231	INFO 201	...	[main] o.s.a.r.c.CachingConnectionFactory : Attempting to connect to: [queue:5672]
worker_1	2021-12-21 22:36:24.260	INFO 201	...	[main] o.s.a.r.c.CachingConnectionFactory : Created new connection: rabbitConnectionFactory#788d9139:0/SimpleConnection@7054c068 [delegate=amqp://guest@172.21.0.2:5672/, localPort= 60394
	]			
worker_1	2021-12-21 22:36:24.320	INFO 201	...	[main] lab.imrec.injector.InjectorApplication : Started InjectorApplication in 6.755 seconds (JVM running for 7.419)

Рис. 5.1 – Приклад роботи контейнеризованого додатку

## 6. Висновки

У ході лабораторної роботи було розвинуто архітектуру попередніх лабораторних робіт, додано додатковий функціонал та контейнеризацію до додатку.