

Kafka를 이용한 E-commerce 애플리케이션 개발 및 구현

정민균¹

¹단국대학교 컴퓨터공학과
32194061@dankook.ac.kr

Abstract

2024학년도 단국대학교 컴퓨터공학과와 고급프로그래밍실습 과제에 대한 보고서이다. 개발 시 많이 쓰이는 플랫폼/프레임워크/기법 중 메시지 큐, REST API, gRPC, MySQL, MongoDB/Neo4j의 위 5가지 기술을 각각 이용해 간단한 애플리케이션 개발이 목표이다. 본 보고서는 5가지 기술 중 메시지 큐 내에서 Kafka를 이용한 애플리케이션에 대한 내용으로 JDK를 이용한 Apache Kafka와 Python을 이용해 E-commerce 시스템을 구현해보았다.

1. Introduction

네이버 쇼핑, 쿠팡 등 다양한 플랫폼에서 인터넷 쇼핑을 사업화하고 있다. 이커머스(e-commerce)는 인터넷이나 네트워크, 디지털 기술을 이용해 상품과 서비스를 사고 파는 전자상거래를 의미한다. 광고, 마케팅, 고객 지원, 배송, 지불 등의 활동도 포함된다. 사용자의 구매 데이터를 바탕으로 주문 내역, 재고, 결제 등 다양한 형태의 기능을 하는 시스템이 동작한다. 본 프로젝트에서는 Kafka를 이용해 고객이 broker에 구매 내용을 publish하면 각자의 기능을 하는 consumer가 broker에서 topic을 subscribe하여 기능을 수행하는 간단한 애플리케이션 기능을 구현해본다.

2. Kafka 이용 방안

Apache Kafka는 실시간으로 스트리밍 데이터를 수집하고 처리하는 데 최적화된 분산 데이터 스토어이다. 스트리밍 데이터는 연속적으로 생성되는 데이터로, 데이터 레코드를 동시에 전송한다. 스트리밍 플랫폼은 이러한 지속적인 데이터 유입을 처리하고 데이터를 순차적이고 점진적으로 처리해야 한다. Producer가 고객의 구매 정보를 지속적으로 kafka broker에 publish하면 구매 내역, 재고 관리, 결제 처리의 역할을 하는 세 개의 Consumer가 이를 각각 순차적으로 처리한다.

- Publisher_1과 _2에서 구매 데이터가 생성
- kafka bootstrap 서버('localhost:9092')에 orders라는 topic으로 publish.
- 구매 내역, 재고 관리, 결제 처리의 역할을 하는 세 개의 Consumer가 orders topic을 subscribe
- subscribe한 데이터를 바탕으로 각자의 기능을 수행.

2.1. Producer

Producer 고객을 담당한다. 구매 내용을 랜덤으로 생성하고 전송한다. Publisher는 1, 2로 구성되어 있으며 둘은 같은 역할을 수행한다. 한 Producer당 4명의 고객에 대한 데이터가 랜덤으로

생성되며 각 프로세스가 순차적으로 broker에 publish된다. Producer 내에 KafkaProducer라는 메소드를 이용해 Bootstrap 서버로 Kafka 브로커의 주소를 설정한다. 구매 데이터를 1초마다 생성하며 데이터를 JSON 형태로 직렬화하여 broker에 전송한다.

2.2. Consumer

Consumer는 E-commerce 시스템을 담당한다. Kafka 서버에 올라온 topic에 대한 데이터를 subscribe 한다. consumer는 구매 내역, 재고 관리, 결제 처리로 각 기능에 따라 세 가지로 각자 동작하는 프로세스이다. KafkaConsumer로 topic과 bootstrap server를 입력받고 subscribe한 데이터를 각자의 기능으로 처리한 뒤 출력한다.

3. 설치 및 실행 방법

패키지 업데이트 및 설치와 tmux를 통해 2개의 Producer와 3개의 Consumer를 실행하는 스크립트를 실행한다. Kafka bootstrap server와 zookeeper를 비롯한 모든 서버와 클라이언트를 실행하는 스크립트이다.

- sudo apt-get update
- sudo apt-get install git
- git clone https://github.com/mingun0112/dku_kafka.git
- or unzip dku_kafka.zip
- cd ./dku_kafka
- sh ./start_kafka.sh

3.1. Kafka 설치 및 실행 방법

위 스크립트에 해당 내용이 모두 통합돼 실행된다.

- curl -O https://dlcdn.apache.org/kafka/3.9.0/kafka_2.13-3.9.0.tgz
- tar -xvzf kafka_2.13-3.9.0.tgz
- cd ./dku_kafka/kafka_2.13-3.9.0

- bin/zookeeper-server-start.sh config/zookeeper.properties
- 새로운 shell 실행
- cd ./dku_kafka/kafka_2.13-3.9.0
- bin/kafka-server-start.sh config/server.properties
- cd ./
- python3 producer_1.py
- python3 producer_2.py
- python3 consumer_order.py
- python3 consumer_inventory.py
- python3 consumer_payment.py

3.2. 설치 방법 사진

```

mingun@DESKTOP-UP9LUKD: ~/submit/dku_kafka
mingun@DESKTOP-UP9LUKD:~/submit$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
mingun@DESKTOP-UP9LUKD:~/submit$ sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.11).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
mingun@DESKTOP-UP9LUKD:~/submit$ git clone https://github.com/mingun0112/dku_kafka.git
Cloning into 'dku_kafka'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 21 (delta 8), reused 19 (delta 6), pack-reused 0 (from 0)
Receiving objects: 100% (21/21), done.
Resolving deltas: 100% (8/8), done.
mingun@DESKTOP-UP9LUKD:~/submit$ cd dku_kafka/
mingun@DESKTOP-UP9LUKD:~/submit/dku_kafka$

```

Figure 1: git에서 레포지토리를 클론한 후 디렉토리에 접근하여 패키지를 업데이트한다.

```

curl -O https://dlcdn.apache.org/kafka/3.9.0/kafka_2.13-3.9.0.tgz
tar -xvzf kafka_2.13-3.9.0.tgz
cd kafka_2.13-3.9.0

tmux new-session -d -s kafka-server
tmux new-session -d -s zookeeper

tmux send-keys -t zookeeper "cd ../dku_kafka/kafka_2.13-3.9.0" C-m
tmux send-keys -t zookeeper "bin/zookeeper-server-start.sh config/zookeeper.properties" C-m

tmux send-keys -t kafka-server "cd ../dku_kafka/kafka_2.13-3.9.0" C-m
tmux send-keys -t kafka-server "bin/kafka-server-start.sh config/server.properties" C-m

# cd ../dku_kafka
# cd kafka_2.13-3.9.0
# bin/kafka-topics.sh --create --topic quickstart-events --bootstrap-server localhost:9092

```

Figure 2: 최종 스크립트인 start_kafka.sh에 kafka 설치에 대한 명령어도 포함되어 있으나 만일의 사태에 대비해 kafka를 설치하고 kafka server를 실행하고 zookeeper를 실행하는 방법에 대한 내용이다.

```

mingun@DESKTOP-UP9LUKD: ~/submit/dku_kafka
mingun@DESKTOP-UP9LUKD:~/submit/dku_kafka$ sh start_kafka.sh
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openjdk-11-jdk is already the newest version (11.0.25+9-1ubuntu1~22.04).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Updating system packages...
Installing Python and pip...
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04.1).
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: kafka-python in /home/mingun/.local/lib/python3.10/site-packages (from -r requirements.txt (line 1)) (2.0.2)
Requirement already satisfied: tabulate in /home/mingun/.local/lib/python3.10/site-packages (from -r requirements.txt (line 2)) (0.9.0)
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 116M 100 116M 0 0 58.5M 0 0:00:01 0:00:01 --:-- 58.5M
kafka_2.13-3.9.0/
kafka_2.13-3.9.0/LICENSE

```

Figure 3: sh ./start_kafka.sh를 실행한다.

3.3. 실행 결과 사진

```
kafka_2.13-3.9.0/libs/rocksdbjni-7.9.2.jar
kafka_2.13-3.9.0/libs/kafka-streams-scala_2.13-3.9.0.jar
kafka_2.13-3.9.0/libs/kafka-streams-test-utils-3.9.0.jar
kafka_2.13-3.9.0/libs/kafka-streams-examples-3.9.0.jar
kafka 실행중
[detached (from session kafka)]
mingun@DESKTOP-UP9LUKD:~/submit/dku_kafka$ tmux ls
kafka: 1 windows (created Mon Dec 9 22:55:00 2024)
kafka-server: 1 windows (created Mon Dec 9 22:54:55 2024)
zookeeper: 1 windows (created Mon Dec 9 22:54:55 2024)
mingun@DESKTOP-UP9LUKD:~/submit/dku_kafka$ tmux attach -t kafka
```

Figure 4: tmux에 애플리케이션에 대한 메인 세션(kafka)과 kafka server, zookeeper가 실행되고 있다.

The terminal window is split into two panes. The left pane shows the output of two producers publishing messages to a Kafka topic. The right pane shows the output of three consumers processing these messages.

Producer 1 Messages (Left Pane):

```
{
  "item_id": "C",
  "quantity": 5,
  "price": 3600
},
{
  "item_id": "A",
  "quantity": 2,
  "price": 1200
},
{
  "item_id": "B",
  "quantity": 3,
  "price": 2800
},
{
  "payment_method": "SamsungPay"
}
```

Producer 2 Messages (Left Pane):

```
{
  "item_id": "B",
  "quantity": 4,
  "price": 2800
},
{
  "item_id": "C",
  "quantity": 4,
  "price": 3600
},
{
  "item_id": "A",
  "quantity": 5,
  "price": 1200
},
{
  "payment_method": "SamsungPay"
}
```

Consumer Logs (Right Pane):

주문 ID: 4555 (이름: 밍)

Item ID	수량	가격
B	4	2800
C	4	3600
A	5	1200

주문 ID: 7752 (이름: Ningina)

Item ID	수량	가격
C	1	3600

주문 ID: 3748 (이름: 대왕)

Item ID	수량	가격
B	5	2800
C	1	2100

주문 내역: 3748
Item B: 잔여량: 2906

현재 재고:

Item ID	재고
A	919
B	2906
C	2100

주문 ID: 1635 : 28900 원
결제 방식: SamsungPay

주문 ID: 4555 : 31600 원
결제 방식: SamsungPay

주문 ID: 7752 : 3600 원
결제 방식: SamsungPay

주문 ID: 3748 : 14000 원
결제 방식: ApplePay

Figure 5: 윈) Producer2개가 하나의 topic으로 publish하고 있다. 우) 3개의 Consumer가 각각의 기능을 수행하고 있다.