

Học phần Quản trị Dữ liệu lớn: Bài thực hành Apache Kafka (3)

Phạm Tiến Lâm, Đặng Văn Báu

Khởi động:

1. Khởi động máy chủ **Zookeeper**.

```
.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
```

2. Khởi động máy chủ **Kafka**.

```
.\bin\windows\kafka-server-start.bat .\config\server.properties
```

3. Tạo **Topic** mới trong môi trường **Kafka**.

Khởi tạo 2 topic mới có tên lần lượt: “topica” và “topicb”

```
kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1  
--partition 1 --topic “topic_name”
```

4. Mở một ‘**producer console**’ cho phép gửi các message tới **Topic** trong Kafka.

```
kafka-console-producer.bat --broker-list localhost:9092 --topic “topic_name”
```

5. Mở một “**consumer console**” và cho phép đọc tin nhắn từ **Topic**.

```
kafka-console-consumer.bat --topic “topic_name” --bootstrap-server  
localhost:9092 --from-beginning
```

6. Kiểm tra các **Topic** đã khởi tạo.

```
kafka-topics.bat --list --bootstrap-server localhost:9092
```

Activity 2: Tích hợp Apache Kafka trong python để đọc và xử lý video

1. Tạo một file producer.py

- Khai báo thư viện

```
import cv2
from kafka import KafkaProducer
import threading
```

- Khởi tạo hàm đọc video và gửi thông tin lên Topic

```
def publish_video_to_kafka(producer, topic, video_path):
    cap = cv2.VideoCapture(video_path)

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        # Convert frame to bytes
        __, buffer = cv2.imencode('.jpg', frame)
        data = buffer.tobytes()

        # Publish frame to Kafka topic
        producer.send(topic, value=data)

    cap.release()
```

- Khởi tạo hàm main():

Cấu hình Kafka Producer; sau đó tạo và chạy hai Kafka Producer cho 2 topic khác nhau trong các luồng riêng biệt.

- + Tạo một luồng “**producer_thread1**”, luồng này sẽ gửi **video_path1** lên topic “**topica**”
- + Tạo một luồng “**producer_threadb**”, luồng này sẽ gửi **video_path2** lên topic “**topicb**”

```
def main():
    # Configure Kafka producer
    bootstrap_servers = 'localhost:9092'
    producer = KafkaProducer(
        bootstrap_servers=['localhost:9092'],
        api_version=(0, 10, 1)
    )

    # Create and run Kafka producer for topic 1 in a separate thread
    topic1 = 'topica'
    video_path1 = 'roads.mp4'
    producer_thread1 = threading.Thread(target=publish_video_to_kafka,
                                         args=(producer, topic1, video_path1))
    producer_thread1.start()

    # Create and run Kafka producer for topic 2 in a separate thread
    topic2 = 'topicb'
    video_path2 = 'roads.mp4'
    producer_thread2 = threading.Thread(target=publish_video_to_kafka,
                                         args=(producer, topic2, video_path2))
    producer_thread2.start()

    producer_thread1.join()
    producer_thread2.join()

if __name__ == "__main__":
    main()
```

2. Tạo một file consumer.py để xử lý các luồng dữ liệu

- Khai báo thư viện

```
import cv2
from kafka import KafkaProducer, KafkaConsumer
import numpy as np
import threading
```

- Khởi tạo hàm “consume_video_from_kafka”: Đọc dữ liệu được gửi từ “topica” và yêu cầu hiển thị ra màn hình video gốc.

```
def consume_video_from_kafka(consumer, topic):
    for message in consumer:
        frame_data = np.frombuffer(message.value, dtype=np.uint8)
        frame = cv2.imdecode(frame_data, cv2.IMREAD_COLOR)
        # gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)

        # Display video with the window name as the topic
        cv2.imshow(topic, frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
```

- Khởi tạo hàm “consume_video_from_kafka2”: Đọc dữ liệu được gửi từ “topicb” và xử lý các khung hình sao cho tách các vật thể ra khỏi khung hình. Cuối cùng yêu cầu hiển thị ra màn hình video sau khi xử lý

```
def consume_video_from_kafka2(consumer, topic):
    last_gray = None
    for message in consumer:
        frame_data = np.frombuffer(message.value, dtype=np.uint8)
        frame = cv2.imdecode(frame_data, cv2.IMREAD_COLOR)
        gray_frame = cv2.GaussianBlur(frame, (5, 5), 0)

        if last_gray is None:
            last_gray = gray_frame
            continue

        diff = cv2.absdiff(gray_frame, last_gray)

        _, difference = cv2.threshold(diff, 25, 255, cv2.THRESH_BINARY)

        last_gray = gray_frame

        # Display video with the window name as the topic
        cv2.imshow(topic, difference)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
```

- Khởi tạo hàm **main**:

Tạo Kafka consumer cho “**topica**” và “**topicb**”

Tạo 2 luồng, mỗi luồng nhận và xử lý dữ liệu từ các frame đồng thời từ consumer và topic được gọi đến.

```
def main():

    topic1 = 'topica'
    topic2 = 'topicb'

    # Create Kafka consumer for topic 1
    consumer1 = KafkaConsumer(
        topic1,
        bootstrap_servers=['localhost:9092'],
        api_version=(0, 10)
    )

    # Create Kafka consumer for topic 2
    consumer2 = KafkaConsumer(
        topic2,
        bootstrap_servers=['localhost:9092'],
        api_version=(0, 10)
    )

    # Create and start threads for consuming videos
    consumer_thread1 = threading.Thread(target=consume_video_from_kafka,
                                         args=(consumer1, topic1))
    consumer_thread2 = threading.Thread(target=consume_video_from_kafka2,
                                         args=(consumer2, topic2))

    consumer_thread1.start()
    consumer_thread2.start()

    # Wait for threads to finish

    consumer_thread1.join()
    consumer_thread2.join()

if __name__ == "__main__":
    main()
```

Bài tập:

Yêu cầu:

1. Chạy file **producer.py** và **consumer.py**
2. Tìm hiểu thêm các kỹ thuật xử lý ảnh trong opencv sao cho xác định vị trí vật thể (xe ô tô) trong “**topicb**”.
(*Contours và bounding box*)
3. Sử dụng kết quả vị trí của các vật thể tìm được trong yêu cầu 2 để đếm số xe đi qua từng làn đường. Lưu trữ số liệu các vật thể xuất hiện trong khoảng giám sát vào file **.txt** (*Yêu cầu thêm*)