KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

CSC17106 – XỬ LÝ PHÂN TÍCH DỮ LIỆU TRỰC TUYẾN

HƯỚNG DẪN THỰC HÀNH

KAFKA CƠ BẢN

I. Thông tin chung

Mã số:	HD02
Thời lượng dự kiến:	3 tiếng
Deadline nộp bài:	-
Hình thức:	-
Hình thức nộp bài:	-
GV phụ trách:	Phạm Minh Tú
Thông tin liên lạc với GV:	pmtu@fit.hcmus.edu.vn

II. Chuẩn đầu ra cần đạt

Bài hướng dẫn này nhằm mục tiêu đạt giúp sinh viên được các mục tiêu sau:

1. Cài đặt Kafka trên môi trường window
2. Các lệnh cơ bản trên Kafka
3. Sử dụng producer và consumer với Python

III. Mô tả

Giới thiệu Kafka

Kafka là một hệ thống xử lý dòng dữ liệu mã nguồn mở và phân phối dựa trên cụm. Nó được tạo ra để xử lý và quản lý dữ liệu luồng lớn và có khả năng mở rộng tốt. Dưới đây là một số khái niệm cơ bản và điểm mạnh của Kafka:

Dòng Dữ Liệu (Data Stream): Kafka là một hệ thống được thiết kế để xử lý dữ liệu dạng luồng (streaming data), tức là dữ liệu được tạo ra và gửi đi liên tục theo thời gian, ví dụ như thông điệp từ cảm biến, log hệ thống, dữ liệu tài chính, và nhiều nguồn dữ liệu khác.

Chủ Đề (Topic): Trong Kafka, dữ liệu được tổ chức thành các chủ đề (topic). Mỗi chủ đề đại diện cho một loại thông điệp hoặc dòng dữ liệu cụ thể. Các producer gửi thông điệp vào các chủ đề, và các consumer đọc thông điệp từ các chủ đề.

Producer: Producer là thành phần tạo ra và gửi thông điệp (message) vào Kafka. Producer đưa thông điệp vào một chủ đề cụ thể và Kafka sẽ quản lý việc phân phối thông điệp này đến các consumer.

Consumer: Consumer là thành phần đọc và xử lý thông điệp từ Kafka. Consumer có thể đăng ký để đọc thông điệp từ một hoặc nhiều chủ đề và thực hiện xử lý dữ liệu như lưu trữ hoặc hiển thị.

Broker: Broker là một máy chủ Kafka chạy trên mạng. Kafka thường được triển khai dưới dạng một cụm (cluster) của các broker để cung cấp khả năng mở rộng và sự đảm bảo dự phòng. Mỗi broker trong cụm có vai trò quản lý một phần của dữ liệu và xử lý yêu cầu từ producer và consumer.

Bảo Mật và Quyền Truy Cập: Kafka cung cấp các tùy chọn bảo mật và kiểm soát quyền truy cập, cho phép bạn đảm bảo rằng dữ liệu của bạn được bảo vệ và chỉ được truy cập bởi những người cần thiết.

Khả Năng Mở Rộng: Kafka được thiết kế để có khả năng mở rộng dễ dàng. Bạn có thể thêm các broker mới vào cụm Kafka để tăng khả năng xử lý và lưu trữ dữ liệu khi cần.

Đảm Bảo Dự Phòng: Kafka đảm bảo dự phòng thông qua việc sao lưu thông điệp và dữ liệu trên nhiều broker khác nhau. Điều này giúp đảm bảo tính sẵn sàng và không bị mất dữ liệu trong trường hợp một số broker gặp sự cố.

Kafka được sử dụng rộng rãi trong các trường hợp sử dụng như quản lý log hệ thống, **xử lý dữ liệu thời gian thực**, lưu trữ và phân phối dữ liệu, và nhiều ứng dụng khác. Nó là một trong những công cụ quan trọng cho xử lý dữ liệu lớn và dòng dữ liệu trong thế giới công nghệ hiện đại.

1. Các bước cài đặt Kafka trên môi trường Window

Bước 1: Tải và Cài Đặt Java

Apache Kafka yêu cầu Java để chạy. Bạn cần cài đặt JDK (Java Development Kit) trước. Hãy tải JDK từ trang chính thức của Oracle hoặc sử dụng OpenJDK.

Bước 2: Tải Apache Kafka

Truy cập trang chính thức của Apache Kafka (https://kafka.apache.org/downloads) và tải phiên bản mới nhất của Kafka về máy tính của bạn. Chọn phiên bản tương ứng với phiên bản Java bạn đã cài đặt.

Bước 3: Giải Nén Apache Kafka

Sau khi tải xong, giải nén tệp tải về vào một thư mục bất kỳ trên máy tính của bạn. Ví dụ, bạn có thể giải nén vào C:\kafka.

Bước 4: Cấu Hình Kafka

Trong thư mục bạn vừa giải nén, bạn sẽ tìm thấy một tệp cấu hình có tên server.properties trong thư mục config. Bạn có thể chỉnh sửa các cấu hình của Kafka trong tệp này nếu cần thiết.

Bước 5: Khởi Động ZooKeeper

Kafka yêu cầu ZooKeeper để quản lý trạng thái và tọa độ của các broker. Bạn cần khởi động ZooKeeper trước khi khởi động Kafka. Bạn có thể tìm tệp zookeeper-server-start.bat trong thư mục bin của Kafka và chạy nó.

C:\kafka\bin\windows\zookeeper-server-start.bat C:\kafka\config\zookeeper.properties

Bước 6: Khởi Động Kafka Broker

Sau khi ZooKeeper đã khởi động, bạn có thể khởi động Kafka broker. Tìm tệp kafka-server-start.bat trong thư mục bin của Kafka và chạy nó.

C:\kafka\bin\windows\kafka-server-start.bat C:\kafka\config\server.properties

Các bước cấu hình kafka chi tiết:

Cấu hình Kafka Broker và Cổng lắng nghe:

```
# Định danh duy nhất của Kafka Broker
broker.id=1

# Cổng Kafka Broker lắng nghe các kết nối
listeners=PLAINTEXT://localhost:9092
```

Cấu hình Thư Mục Lưu Trữ Log

```
# Thư mục lưu trữ log của Kafka
log.dirs=/path/to/kafka-logs
```

Cấu hình ZooKeeper Connection String

```
# Địa chỉ và cổng ZooKeeper
zookeeper.connect=localhost:2181
```

Cấu hình Số Lượng Partitions Mặc Định

```
# Số lượng partitions mặc định khi tạo chủ đề mới
num.partitions=3
```

Cấu hình Số Lượng Replicas Mặc Định

```
# Số lượng replica mặc định khi tạo chủ đề mới
default.replication.factor=2
```

Cấu hình Retention Period và Retention Policy

```
# Số giờ dữ liệu sẽ được giữ lại trước khi bị xóa đi
log.retention.hours=168

# Chế độ làm sạch dữ liệu: 'delete' hoặc 'compact'
log.cleanup.policy=delete
```

Cấu hình Bảo Mật (SSL):

```
# Xác định giao thức bảo mật cho các kết nối
security.inter.broker.protocol=SSL
```

2. Các lệnh cơ bản trên Kafka

Tạo chủ đề (topic) Kafka

C:\kafka\bin\windows\kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic test

Kafka Console Producer

Lệnh kafka-console-producer cho phép bạn tạo producer để sản xuất (publish) các thông điệp vào chủ đề (topic) Kafka. Ví dụ sau đây sẽ tạo một producer và gửi các thông điệp đến chủ đề "test":

kafka-console-producer.sh --broker-list localhost:9092 --topic test

Sau khi chạy lệnh này, bạn có thể nhập các thông điệp từ bàn phím và nhấn Enter để gửi chúng đến chủ đề "test".

Kafka Console Consumer

Lệnh kafka-console-consumer cho phép bạn tạo consumer để tiêu thụ (consume) các thông điệp từ một chủ đề Kafka. Ví dụ sau đây sẽ tạo một consumer để tiêu thụ thông điệp từ chủ đề "test":

kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning

Lệnh --from-beginning sẽ cho phép bạn đọc tất cả các thông điệp đã tồn tại trong chủ đề "test" từ đầu.

3. Sử dụng producer và consumer với Python

Để sử dụng producer và consumer Kafka trong Python, bạn có thể sử dụng thư viện confluent-kafka-python, một thư viện Kafka phổ biến cho Python. Để bắt đầu, bạn cần cài đặt thư viện này bằng pip:

pip install confluent-kafka

Kafka Producer trong Python:

```
from confluent_kafka import Producer

# Cấu hình producer
producer_config = {
    'bootstrap.servers': 127.0.0.1:9092',  # Địa chỉ Kafka broker
    'client.id': 'python-producer'
}

# Khởi tạo producer
producer = Producer(producer_config)

# Chủ đề Kafka mà bạn muốn gửi thông điệp đến
topic = 'test'

# Gửi thông điệp đến chủ đề
```

```
for i in range(10):
    message = f'Message {i}'
    producer.produce(topic, key=str(i), value=message)
    // Sleep khoảng 1 giây (sinh viên tự code)

# Chờ cho đến khi tất cả các thông điệp đã được gửi
producer.flush()
print('Messages sent successfully')
```

Kafka Consumer trong Python:

```
from confluent_kafka import Consumer, KafkaError

# Cấu hình consumer
consumer_config = {
    'bootstrap.servers': 'localhost:9092',  # Địa chỉ Kafka broker
    'group.id': 'my-group',
    'auto.offset.reset': 'earliest'  # Đọc từ đầu
}

# Khởi tạo consumer
consumer = Consumer(consumer_config)

# Chủ đề Kafka mà bạn muốn tiêu thụ thông điệp
topic = 'test'

# Đăng ký chủ đề
consumer.subscribe([topic])

# Tiêu thụ thông điệp từ chủ đề
while True:
    msg = consumer.poll(1.0)  # Chờ một giây
    if msg is None:
        continue
    if msg.error():
        if msg.error().code() == KafkaError._PARTITION_EOF:
            print('Reached end of partition')
        else:
            print(f'Error: {msg.error()}\n')
    else:
        print(f'Received message: {msg.value().decode("utf-8")}')

# Đóng consumer
consumer.close()
```

IV. Tài liệu tham khảo

1. https://kafka.apache.org/
2. https://github.com/apache/kafka
3. https://docs.confluent.io/platform/current/index.html
4. https://docs.confluent.io/tutorials/
5. https://kafka.apache.org/quickstart
6. https://stackoverflow.com/questions/tagged/kafka
7. https://www.linkedin.com/groups/12306331/

V. Bài tập

Mô tả:

Bạn là một nhà quản trị mạng và cần tạo một ứng dụng để theo dõi trạng thái mạng của các thiết bị mạng trong thời gian thực. Bài tập này yêu cầu bạn tạo một hệ thống xử lý dữ liệu thời gian thực sử dụng Apache Kafka để gửi và nhận thông điệp trạng thái mạng.

Yêu Cầu:

1. Tạo một producer Kafka Python để gửi thông điệp trạng thái mạng từ các thiết bị mạng (ví dụ: máy chủ, router, switch) vào chủ đề Kafka.
2. Tạo một consumer Kafka Python để đọc thông điệp trạng thái mạng từ chủ đề Kafka và hiển thị trạng thái mạng lên màn hình.
3. Thực hiện xử lý trạng thái mạng trong consumer, ví dụ: kiểm tra nếu trạng thái mạng là "Online" hoặc "Offline" và lưu trạng thái mạng hiện tại.

Yêu cầu nâng cao:

Bài 1:

Ngữ cảnh thực tế:

- Một công ty có nhiều ứng dụng web chạy trên các server khác nhau. Cần một hệ thống tập trung để thu thập và xử lý log.
- Log bao gồm: `log_level`, `service_name`, `message`, `timestamp`.

Yêu cầu:

1. **Producer:**
   a. Giả lập các ứng dụng gửi log với các mức độ: `INFO`, `ERROR`, `DEBUG`.
   b. Gửi vào topic `app-logs`.
2. **Consumer:**
   a. Tách riêng các log `ERROR` để xử lý ưu tiên.
   b. Ghi log lỗi vào file `error.log`.
   c. Tính toán và in ra số lượng log theo từng mức độ sau mỗi 10 giây.

Gợi ý:

- Cách sử dụng **Kafka Headers** để gắn metadata (mức độ log).
- Cấu hình **consumer group** để xử lý log song song.
- Tìm hiểu về **log compaction** để lưu trữ log hiệu quả.

Bài 2:

Ngữ cảnh thực tế:

- Một ứng dụng di động muốn phân tích hành vi người dùng dựa trên các sự kiện như: `click`, `view`, `purchase`.

Yêu cầu:

1. **Producer:**

   a. Gửi sự kiện với các trường:
      i. `user_id`, `event_type`, `item_id`, `timestamp`.
   b. Sử dụng **Custom Partitioner** để phân loại:
      i. `click` vào partition 0.
      ii. `view` vào partition 1.
      iii. `purchase` vào partition 2.

2. **Consumer:**
   a. Đọc và đếm số lượng `click`, `view`, `purchase` mỗi phút.
   b. Ghi log kết quả.

Gợi ý tự tìm hiểu:

- Cách viết **Custom Partitioner** cho Producer.
- Cách cấu hình và kiểm tra các **partitions** trong Kafka.
- Tài liệu về **Kafka Internals** để hiểu rõ cách dữ liệu được phân phối.