

Apache Kafka를 활용한 실시간 대규모 비디오 스트리밍 기법

유연준¹, 홍석민², 신용태³

¹숭실대학교 컴퓨터학과 석사과정

²숭실대학교 컴퓨터학과 석사과정

³숭실대학교 컴퓨터학과 교수

dbduswns1997@soongsil.ac.kr, ghdtjrals3@soongsil.ac.kr, shin@ssu.ac.kr

Real-time Watermarking Method for Streaming Video Data

Yeon-Jun Yoo¹, Seok-Min Hong², Yong-Tae Shin³

¹Dept. of Computer Science, Soongsil University

²Dept. of Computer Science, Soongsil University

³Dept. of Computer Science, Soongsil University

요 약

오늘날 클라우드 컴퓨팅은 FIFA, WTA, F1, MLB 등과 같은 비디오 및 실시간 스포츠 이벤트에 널리 사용된다. DataM에 따르면 비디오 스트리밍 플랫폼 시장은 545억 달러에서 2,523달러에 달할 것으로 예측된다. 기존 실시간 스트리밍 방법은 스트리밍 비디오의 개수가 증가하거나 스트리밍 이용자가 증가할 경우 성능 저하 문제가 발생한다. 본 논문에서는 Apache Kafka Server를 활용한 대규모 비디오 스트리밍 기법을 제안한다. Apache Kafka Server를 사용하여 네트워크를 수집하면 대규모 데이터를 처리할 수 있으며, 데이터의 안정성과 실시간 처리를 할 수 있어 온라인 비디오 스트리밍에 적합하다. 이에 비디오 품질을 선택할 때 적합한 비디오 품질을 선택할 수 있다. 향후 제안하는 기법은 많은 데이터와 실험으로 실질적인 검증을 할 예정이다.

1. 서론

클라우드 컴퓨팅은 오늘날 FIFA, WTA, F1, MLB 등과 같은 비디오 스트리밍 및 실시간 스트리밍 스포츠 이벤트에 널리 사용된다. 또한 DataM에 따르면 세계 실시간 비디오 스트리밍 플랫폼 시장은 2022년 545억 달러에서 2030년 2,523억 달러에 달할 것으로 예측한다. 실시간 스트리밍은 시청자와의 독특한 상호작용과 참여를 제공한다. 크리에이터는 실시간으로 댓글과 질문에 답변할 수 있으며, 커뮤니티와 유대감을 형성할 수 있다.

기존 실시간 스트리밍은 크게 세 가지 과정을 거쳐서 사용자에게 영상이 전달된다. 세 가지 과정은 트랜스코딩, 패키징, CDN(Content Delivery Network) 저장으로 이루어져 있다. 트랜스코딩 과정은 원본 영상 파일을 인코딩하여, 여러 해상도와 코덱을 가진 영상 파일을 만드는 과정이다. 패키징은 영상 파일을 몇 초 단위로 잘라 스트리밍에 용이한 구조를 가진 파일을 만든다. CDN 저장 과정은 만들어진 영상 파일을 CDN에 올리는 과정이다. 위의 과정을 통

해 실시간 스트리밍이 가능해진다. 하지만 기존 실시간 스트리밍 방법은 스트리밍 비디오의 개수가 증가하거나 스트리밍 이용자가 증가할 경우 성능 저하 문제가 발생한다.

이에 본 논문에서는 Apache Kafka를 활용한 실시간 대규모 비디오 스트리밍 기법을 제안한다. 제안하는 기법은 Apache Kafka를 기반 서버를 통해 사용자의 장치에서 생성되는 네트워크 및 성능 데이터를 수집한다. 수집된 데이터는 Storage에 저장된다. HAS(Http Adaptive Streaming)를 통해 여러 해상도의 영상이 생성되며 네트워크 트래픽과 영상을 Rate-based 알고리즘을 통해 적합한 해상도의 영상을 선택하여 전송한다. 위의 과정을 통해 사용자가 비디오 시청 시 QoE(Quality of Experience)를 높일 수 있다.

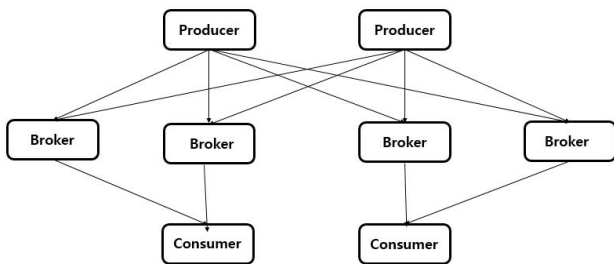
본 논문의 구성은 다음과 같다. 2장에서는 Apache Kafka와 HAS(Http Adaptive Streaming)에 대해 살펴보고, 3장에서는 본 논문에서 제안하는 기법을 설명하며, 4장에서는 결론 및 향후 계획을 제시한다.

2. 관련연구

본 장에서는 제안하는 기법에 활용하는 Apache Kafka와 HAS(Http Adaptive Streaming)에 대해 살펴본다.

2.1 Apache Kafka

Apache Kafka는 분산 스트리밍 플랫폼으로, 대용량의 실시간 데이터 스트리밍을 처리하는 데 사용된다. Pub/Sub 모델의 메시지 큐 형태로 동작하며 분산 환경에 특화되어 있다. 아래 그림[1]은 Apache Kafka의 구성도이다.

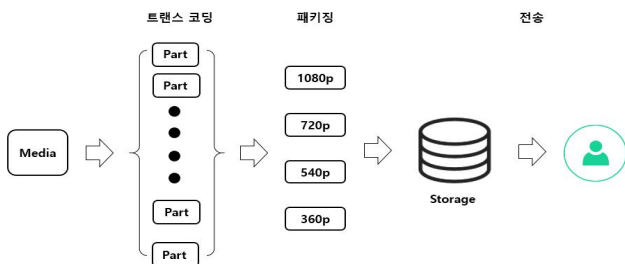


(그림 1) Apache Kafka 구성도

Broker는 클러스터 안에 구성된 여러 서버 중 각 서버를 의미하며 레코드 형태인 메시지의 저장과 검색을 전달하고 관리한다. Producer Topic에 레코드를 전송 또는 생성하는 역할을 한다. Consumer는 Topic에서 레코드를 읽는다. 하나 이상의 Topic을 구독하고, Broker로부터 레코드를 소비한다. Topic은 Producer로 전송된 레코드의 카테고리이다.

2.2 HAS(Http Adaptive Streaming)

HAS는 온라인 비디오 스트리밍을 위한 기술 중 하나로, 동적으로 네트워크 대역폭과 장치의 처리능력에 따라 비디오 품질을 조정하여 최적의 시청 경험을 제공하는 방식이다. HAS의 작동 방식은 다음 그림[2]과 같다.



(그림 2) HAS 작동 방식

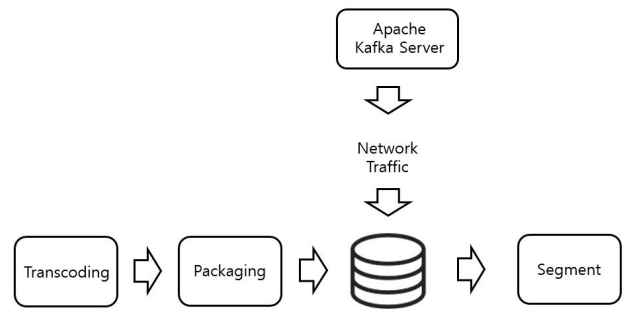
HAS 작동 과정은 크게 3가지로 나누어진다. 트랜스코딩 과정은 원본 영상을 인코딩을 통해 여러 해상도를 가진 여러 별의 영상 파일을 만들어내는 과정이다. 패키징은 여러 별의 영상을 몇 초 단위로 쪼개서 독립적으로 재생이 가능한 단위인 Segment한 후, Storage에 저장하는 과정이다. 후에 사용자의 네트워크 성능에 알맞은 영상을 사용자에게 전송한다.

3. 제안하는 기법

본 장에서는 Apache Kafka를 이용한 실시간 비디오 스트리밍 방법을 설명한다.

3.1 제안하는 기법

제안하는 기법은 Apache Kafka Server를 이용하여 이용자의 네트워크 트래픽을 수집하고 이를 Storage에 전송하여 대역폭과 영상의 품질을 계산한 후, 이용자에게 적절한 품질의 영상을 전송한다. 아래 그림은 제안하는 기법의 순서도[그림3]이다.



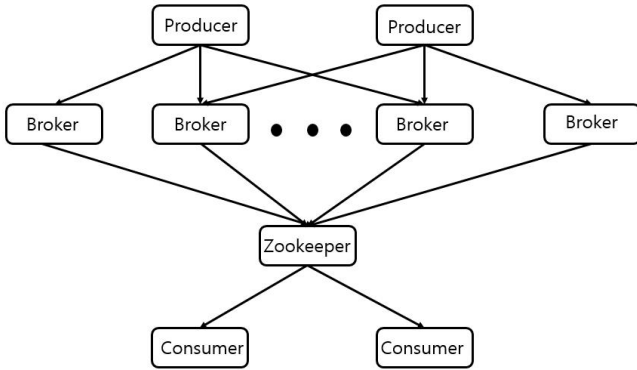
(그림 3) 제안하는 기법 순서도

트랜스코딩과 패키징을 거쳐 여러 해상도를 가진 여러 별의 영상 파일이 Storage에 저장된다. 이후 이용자가 동영상을 시청할 때 Apache Kafka Server에서 이용자의 네트워크 트래픽을 수집한다. 수집된 네트워크 정보와 영상의 비트레이트를 Rate-based 알고리즘을 활용하여 적합한 해상도의 영상을 선택하여 전송한다.

3.2 Apache Kafka Server

제안하는 Apache Kafka Server는 네트워크 트래픽을 수집하고 전달한다. 수집된 네트워크 트래픽은 Producer로 전송된다. 전송된 트래픽은 브로커를 통해 파티션으로 저장된다. Kafka 클러스터를 관리하여 데이터를 안정적으로 저장하고 전송하기 위해

Zookeeper를 사용한다. 아래 그림[4]은 제안하는 Apache Kafka Server이다.



(그림 4) 제안하는 Apache Kafak Server

수집된 네트워크 트래픽은 안정적으로 Storage에 전달된다.

4. 결론.

제안하는 기법은 Apache Kafka Server를 사용하여 온라인 비디오 스트리밍에서 적합한 비디오 품질을 선택하는 기법이다. Apache Kafka Server를 사용하여 네트워크를 수집하면 대규모 데이터를 처리할 수 있으며, 데이터의 안정성과 실시간 처리를 할 수 있어 온라인 비디오 스트리밍에 적합하다. 이에 비디오 품질을 선택할 때 적합한 비디오 품질을 선택할 수 있다.

향후 본 연구에서 제안하는 기법의 많은 데이터와 실험으로 실질적인 검증을 할 예정이다.

“본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터사업의 연구결과로 수행되었음”(IITP-2024-2020-0-01602)

참고문헌

- [1] Roger S. Pressman "Software Engineering A Practltiners' Approach" 3rd Ed. McGraw Hill
- [2] Deleuze, Gilles, Félix Guattari, and Slavica Miletic. Kafka. Literarno-umetniško društvo Literatura, 1995.
- [3]Kreps, Jay, Neha Narkhede, and Jun Rao. "Kafka: A distributed messaging system for log processing." Proceedings of the NetDB. Vol. 11. No. 2011. 2011.
- [4]Summers, Jim, et al. "Characterizing the workload of a Netflix streaming video server." 2016 IEEE International Symposium on Workload Characterization (IISWC). IEEE, 2016.