

# 클라우드 기반 UHD 영상 트랜스코딩 시스템

문희철, 김용환, 김동혁  
 전자부품연구원 멀티미디어 IP 연구센터  
 hcmoon@keti.re.kr, yonghwan@keti.re.kr, kdh007@keti.re.kr

## UHD Video Transcoding System in Cloud Computing Environment

Hee-Cheol Moon, Yong-Hwan Kim, Dong-Hyeok Kim  
 Multimedia IP Research Center Korea Electronics Technology Institute

### 요 약

UHD 영상 콘텐츠는 FHD 영상에 비해 생생하고 더 좋은 고화질의 영상을 제공하지만 영상정보의 데이터 양은 4K UHD 경우 4 배 이상이다. 이러한 초대용량의 UHD 영상을 기존의 병렬/분산 처리를 이용하여 비디오 코딩 한다면 UHD 의 초대용량 특성으로 인하여 연산량 부하가 발생하게 된다. 따라서 UHD 영상은 기존의 분산처리 방식이 아닌 초대용량 데이터를 빠르게 처리 할 수 있는 새로운 분산 처리기술이 필요하다. 본 논문은 UHD 콘텐츠를 빠르게 트랜스코딩 할 수 있는 클라우드 기반 UHD 영상 트랜스코딩 시스템을 제안한다. 본 논문에서 제안하는 UHD 영상 트랜스코딩 시스템은 다음 3 가지 패키지 분석기, 분산 트랜스코더, 스트림 합성기로 구성된다. 패키지 분석기는 입력 영상을 분석하여 오디오와 비디오 스트림을 분할하고 비디오 스트림은 분산 처리를 할 수 있도록 영상 패킷을 분할한다. 분산 트랜스코더는 클라우드 환경을 이용하여 분할된 영상 패킷들을 분산 디코드 및 인코드 처리한다. 스트림 합성기는 트랜스코딩이 완료된 비디오 스트림과 패키지 분석기에서 획득하였던 오디오 스트림을 합성하는 기능을 한다. 제시하는 방안을 적용하여 클라우드 기반 영상 트랜스 코딩 시스템을 구현하였으며, 구현된 시스템은 대용량의 UHD 영상을 빠른 속도로 트랜스코딩이 가능하다.

### 1. 서론

최근 4/16K UHD(Ultra High Definition) 콘텐츠를 지원하는 스마트 TV 나 모바일 기기 등이 시장에 나오면서 UHD 관련 서비스에 대한 관심이 높아지고 있다. 하지만 4K UHD 콘텐츠의 경우 데이터양이 1080P FULL HD 콘텐츠보다 4 배 이상이고, 압축하지 않았을 경우에는 프레임당 25MB 정도의 데이터 크기를 가진다. 이러한 이유로 UHD 콘텐츠를 인터넷에서 스트리밍 하기 위해서는 HEVC(High Efficiency Video Coding)기반 초고효율 코덱 기술이 필수적이다. 하지만 HEVC 는 기존 MPEG-4 AVC|H.264 대비 압축률이 두 배 가까이 높아졌다는 장점이 있지만, 복잡도가 3 배 이상 높아져 연산량 증가에 따른 부하가 발생한다. 따라서 빠른 연산을 위해서 병렬/분산 처리 기술이 필요하게 된다. 또한, N-스크린 환경에서 SD/HD/UHD 콘텐츠 소비가 공존할 것으로 예측되므로 다양한 해상도를 동시 지원하는 초고효율 트랜스코더 기술이 요구된다.

현재 국내외에서 병렬/분산 처리를 이용한 비디오 코딩 기술이 개발되고 연구되었지만, UHD 콘텐츠를 지원하는 시스템은 아직 미흡하다. UHD 의 초대용량 특성으로 인해 기존의 병렬/분산처리 방식으로는 비디오 코딩이 원활히 이루어질 수 없기 때문이다. 따라서 이러한 문제점들을 극복하기 위해 본 연구에서는 클라우드 환경을 이용한 분산 트랜스코딩 시스템을 제안한다.

논문의 구성은 다음과 같다. 2 장에서는 본 논문에서 제안하는 분산 트랜스코딩 시스템에 대해 설명하고, 3 장에서는 제안한 방안을 이용한 구현 및 사례를 설명한다. 마지막으로 4 장에서는 결론을 맺는다.

### 2. 분산 트랜스코딩 시스템

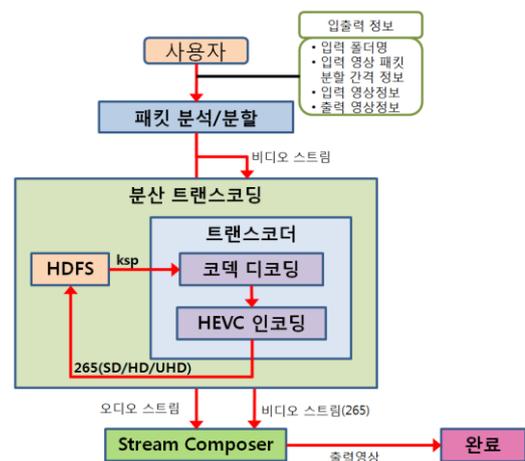


그림 1. 분산 트랜스코딩 시스템 구조

그림 1 은 본 논문에서 제안하는 분산 트랜스코딩 시스템의 전체 구조를 나타낸다. 사용자는 AVI, MP4, MXF 등의 파일포맷 동영상 데이터를 업로드 한 후 트랜스코딩 명령을 실행하면 패킷 분석기는 입력데이터를 분석하여 영상과 음성 스트림으로 분할한다. 음성 스트림은 스트림 합성기로 전송되고 영상 스트림은 분산처리를 하기 위해 GOP(Group of Pictures) 단위로 패킷을 분할 한다. 분할된 패킷들은 클라우드 파일 시스템에 저장되고 이 패킷들을 분산 트랜스코더에서 코덱 포맷에 맞게 디코딩/인코딩을 수행한 후 다시 클라우드 파일 시스템에 저장한다. 마지막 스트림 합성기는 분산 트랜스코딩된 패킷들을 하나의 스트림으로 합성하고 패킷 분석기에 분리되었던 음성 스트림을 추가하여 MP4 및 MXF 파일 포맷으로 변환 후 시스템을 종료한다.

### 2.1 패킷 분석기

패킷 분석기는 다양한 포맷의 비디오 패킷을 분석하고 분할하는 기능을 한다. 비디오 영상이 입력되면 영상의 코덱 정보를 파악하여 입력 비디오 데이터의 스트림을 일정 시간 단위로 패킷을 분할하고 클라우드 파일시스템에 저장한다.

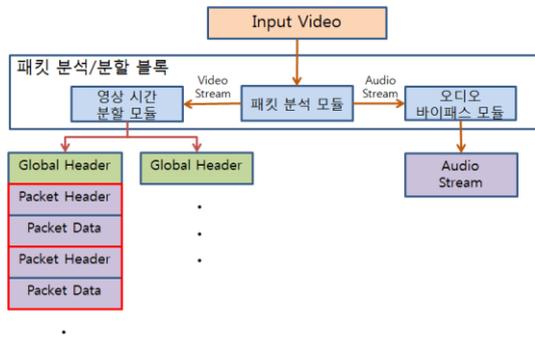


그림 2. 패킷 분석기

그림 3 에서 보면 하나의 분할 스트림 내에는 하나의 Global Header 와 수십~수백 개의 Video Packet 이 존재하게 된다. Global Header 에는 전역 정보가 포함되고, 각각의 Video Packet 은 Video Packet Header 와 Video Packet Payload 로 구성된다. Video Packet Payload 는 픽처 한 장에 해당되는 압축된 비트스트림이다.

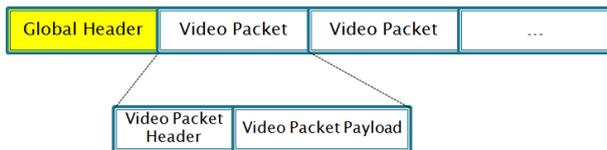


그림 3. 분할 스트림 구조

### 2.2 분산 트랜스코더

그림 4 는 분산 트랜스코더의 구조를 나타낸다. 분산 트랜스코더에서는 클라우드에 저장된 일정 시간분할 영상 패킷들을 디코드/인코드 작업을 수행한다. 본 연구에서는 클라우드 분산처리를 하기 위해 하둡 클라우드 환경을 이용한다. 파일 시스템은 하둡 파일시스템(HDFS, Hadoop Distributed File System)을 이용하였고, 분산처리는 하둡 프레임워크인 MapReduce 를 이용하였다[2,3].

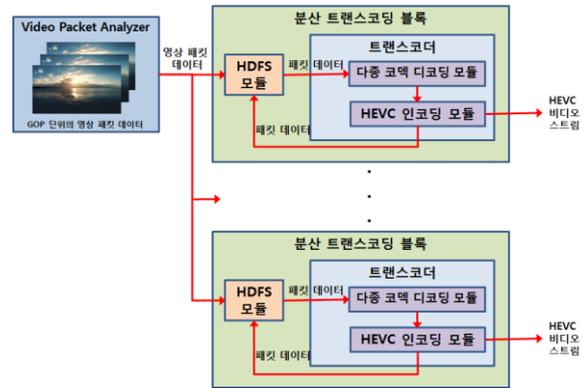


그림 4. 분산 트랜스코더

임워크인 MapReduce 를 이용하였다[2,3].

영상 트랜스코딩을 위해서는 Java 로 구현된 하둡의 MapReduce 와 C++로 구현된 Video Transcoder 의 상호 데이터 전송 과정이 필요하다. 따라서 그림 5 와 같이 JNI(Java Native Interface)를 이용해서 패킷의 Byte Data 를 Video Transcoder 에 전송하여 트랜스코딩을 한 다음, 다시 Byte Data 를 하둡 파일 시스템에 전송한다.

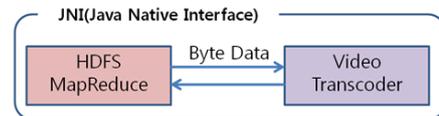


그림 5. 하둡 HDFS 와 트랜스코더 데이터 전송 방법

### 2.3 스트림 합성기

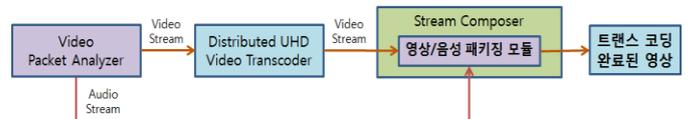


그림 6. 스트림 합성기 관계도

스트림 합성기는 패킷 분석기에서 획득하였던 오디오 스트림과 분산 트랜스코더에서 트랜스코딩이 완료된 영상의 패킷들을 합성하는 기능을 한다. 합성된 영상 스트림과 비디오 스트림은 하나의 동영상 파일(MP4, MXF)로 변환된다. 따라서 스트림 합성기를 통해 사용자가 볼 수 있는 최종 영상이 생성된다. 그림 6 은 스트림 합성기의 오디오와 비디오 스트림 데이터 흐름 관계를 나타내고 있다.

## 3. 구현 및 적용사례

제안된 방안을 이용하여 클라우드 기반 분산 트랜스코딩 시스템을 구축하였다. 클러스터 구성은 Master 1 대, Slave 3 대로 하였으며 하둡 패키지는 Window OS 를 지원해주는 Hortonworks[5]의 HDP(Hortonworks Data Platform) 2.1 을 이용하여 Window 7 에서 클라우드 환경을 구축하였다. 하둡의 버전은 2.4, 분산처리를 지원해주는 MapReduce 의 Map 개수는 Slave Node 당 2 개씩 순차적으로 처리하도록 하였다. 트랜스코더의 디코더 기능은 FFmpeg[4] 라이브러리를 이용하였으며 인코더는 KETI(Korea Electronics Technology

Institute)에서 개발한 HEVC 부호화기를 이용하였다. 그리고 C++로 개발된 트랜스코더와 JAVA 로 개발된 하둡의 MapReduce 의 데이터 전송 인터페이스는 JNI 를 이용하여 구현하였다. 표 1 은 패킷 분할기에서 분할된 패킷 데이터들을 분산 트랜스코더에 처리하여 도출된 결과이다.

● 실행 환경

- ✓OS: Windows 7 SP (64-bit)
- ✓CPU: dual Xeon E5-2650
- ✓RAM: 64 GB
- ✓콘텐츠: 4K(3840x2160) 영상(YCbCr 4:2:0, 8-bit, 60HZ)

표 1. 실험 결과

시퀀스	프레임수	입력 비트율[kbps]	트랜스코딩 속도(fps)	율력 PSNR[dB]	율력 비트율[kbps]
FC안양 축구(스포츠 실외)	4286	41339	10.98	42.41	11701
Jack(CG)	3877	43541	18.86	46.36	5355
도시(야경)	2706	39177	26.29	42.98	6590
식물원(자연 근거리 풍경)	3822	41015	14.13	46.96	4219
안양현라 하키(스포츠 실내)	2988	41722	10.78	47.48	7237
자연풍경 광화문(원거리 풍경)	6237	40869	27.40	40.99	13165
물은날의 초상(유지집)	6963	40620	24.89	52.20	1769
평균	4411	41183	19.05	45.62	7148

입력 영상은 3840x2160 의 해상도를 갖는 UHD 영상 7 개를 이용하였으며, 입력 영상의 평균 프레임 수는 4411 이다. 각각의 영상을 6 번 분산 트랜스코딩을 수행하여 평균 트랜스코딩 속도를 계산하였다. 표 1 에서 보는 바와 같이 7 개의 UHD 영상을 본 논문에서 제안하는 분산 트랜스코더를 적용하면 평균 19.05 fps(frame per second)@45.62dB 로 트랜스코딩을 할 수 있다.

4. 결론

차세대 콘텐츠로 관심을 받고 있는 UHD 콘텐츠는 초고화질을 제공하지만 방대한 데이터를 처리하기 위해 많은 시간과 비용이 요구된다. 본 논문에서는 클라우드 기반 분산 트랜스코딩 시스템을 제시하였으며, 이를 통해 UHD 영상과 같은 대용량 데이터 처리 시 발생하는 스탠드얼론 장비의 처리성능 한계를 극복 할 수 있다.

추후 연구로는 분산 트랜스코더 시스템의 최적화 요소들을 실험을 통해 선별하고자 한다. 패킷 분석기의 패킷 분할 간격, 분산 트랜스코더의 Slave Node 작업 처리 수, 하둡 관련 튜닝항목(블록 복제수, 블록 사이즈, MapReduce 메모리)등을 실험을 통해 UHD 영상 분산 트랜스코딩 속도에 영향을 주는 항목들을 선별하고 최적화된 수치를 얻고자 한다.

감사의 글

This work was supported in part by the IT R&D program of Ministry of Science, ICT and Future Planning, Republic of Korea under Grant No. 1711009199

참고문헌

- [1] 박수현 외 3 명, “ 맵리듀스기반 스트림 데이터 처리를 위한 스트림 할당 및 분할 기법”, 정보과학회논문지, Vol, 19, No 5, pp. 439-443, 2013
- [2] Hadoop, <http://hadoop.apache.org>
- [3] Jeffrey Dean and Sanjay Ghemawat, “ MapReduce: Simplified Data Processing on Large Clusters”, 6th Symposium on Operating System Design and Implementation, 2004
- [4] FFmpeg, <http://ffmpeg.org>
- [5] Hortonworks, <http://hortonworks.com>
- [6] Myoungjin Kim, “ Towards Efficient Design and Implementation of a Hadoop-based Distributed Video Transcoding System in Cloud Computing Environment”, International Journal of Multimedia and Ubiquitous Engineering, Vol. 8, No. 2, pp. 213-224, 2013.03.
- [7] Shanthi.B.R, “ Dynamic Resource Allocation And DistributedVideo Transcoding Using Hadoop CloudComputing”, ICGICT2014, Vol. 2, 2014.03.