

N-Screen 적응형 실시간 트랜스코딩 방법론에 관한 연구

김병수*, 강이철**, 김종우*** 조성웅***

*인천정보산업진흥원 IT융합진흥부

**정보화진흥원 네트워크선도서비스부

***정보통신산업진흥원 IoT사업팀

****BBMC(주) 소프트웨어개발팀

e-mail : kbs2exe@iit.or.kr, kangyc@nia.or.kr, jwkim154@nipa.kr,
swcho@bbmc.co.kr

A Study on Method of Realtime Transcoding For N-Screen Environmenting

Jong-Ryun Lee*, Yi-Chul Kang**, Jong-Woo Kim*** Sung-woong Cho****

*Incheon Information Technology Industry Promotion Agency

**Digital Infrastructure Division, National Information Society Agency

***Dept of IoT, National IT Industry Promotion Agency

****Software Development Team, BBMC Corp

요 약

최근 무선통신의 급격한 발전과 스마트 기기의 확산으로 인해 Tving, pooq 등 다양한 모바일 방송 서비스가 급속도로 증가하고 있다. 또한 다양한 영상처리 기법 등이 등장함에 따라 4K, 8K급의 UHD 동영상이 속속들이 등장하고 있다. 이로 인해 트랜스코딩을 통해 가공되는 동영상의 포맷 및 해상도 또한 매우 다양해질 것으로 전망된다. 현재까지의 트랜스코딩 연구사례는 사용자의 이동환경을 고려한 안정적 QoS 보장 또는 서버의 부하를 줄이기 위한 분산처리 기법 등의 연구 위주로 진행되어 온 것이 현실이다. 하지만 상기 조건(adaptive streaming 및 서버부하 처리)들을 충족시키긴 위해선 보다 효율적인 트랜스코딩 시스템의 제공이 선행 되어야 할 것이다. 이에 따라 본 논문에서는 사용자 관점에서 보다 빨리 스트리밍 서비스를 제공 받기 위하여 우선순위 큐 알고리즘을 적용한 시스템을 설계 및 구현하였다. 검증을 위하여 4가지 컨테이너(.MOV, .FLV, .MKV, .AVI)를 실험대상으로 하였고, 비교 대상 트랜스코딩 시스템은 상용 스트리밍 서비스인 YouTube를 활용하였다. 성능 측정결과, 총 트랜스코딩 완료시간은 YouTube에 비해 41.61%로 시간이 단축되었다. 또한 모바일 TV시청자가 55%를 차지한다는 점을 고려하여 컨트롤 서버에서는 최단시간 서비스 제공을 위하여 저해상도부터 추출하여 스트리밍 서버를 통해 송출하도록 구현하였다. 본 연구결과는 트랜스코딩 성능개선 뿐만 아니라 모바일 대상자를 위한 맞춤형 서비스를 보다 빨리 제공할 수 있을 것이며, 그 수요는 점차 증대될 것으로 예상된다.

1. 서론

최근 무선통신의 급격한 발전과 스마트 기기의 확산으로 인해 Tving, pooq 등 다양한 모바일 방송 서비스가 급속도로 증가하고 있다. 또한 다양한 영상처리 기법 등이 등장함에 따라 4K, 8K급의 UHD 동영상들이 속속들이 등장하고 있다. 이와 같은 대용량 동영상을 끊김없이 전송하기 위해서 단말기 및 통신환경에 맞추어 전송할 수 있도록 다양한 형태로 가공해 주어야 한다.

위와 같은 'One-Source Multi-Use' 개념을 구현하기 위해서는 원래의 콘텐츠를 각 네트워크 대역폭, 단말기 특성 및 성능에 적합하도록 멀티미디어 콘텐츠를 가공/변환/선별하는 과정이 필요하며, 이러한 과정을 멀티미디어 트랜스코딩이라 한다.[1]

트랜스코딩은 로컬에서 수행하는 방법과 서버에서 수행하는 방법이 있는데, 본 논문에서는 서버에서 수행하는

방법만을 다루고자 한다. 논문의 구성은 2장에서는 시스템 구성과 방식에 대하여 설명하고, 3장에서는 구현한 시스템의 성능측정 및 결과를 분석한다. 4장에서는 본 논문의 결론을 맺는다.

2. 본론

2.1 분산형 트랜스코딩

기존의 트랜스코딩은 다수의 트랜스코딩 서버가 존재하더라도 한 개의 콘텐츠에서 산출되어 나오는 로직으로 구성되어 있다. 이는 다수의 콘텐츠를 트랜스코딩함에 있어서 일부 서버에 시간이 많이 걸리는 트랜스코딩이 편중되어 전체 효율을 저하 시키는 현상이 발생한다. 또한 모든 포맷이 변환 완료되어야 게시할 수 있는 단점이 있다. 분산형 트랜스코딩은 각 서버 대기열의 트랜스코딩 용량

을 비교하여 콘텐츠를 분산하여 효율을 높이고, 한 개의 콘텐츠를 분산해서 처리하기 때문에 한가지 파일만 완료한 이후에 콘텐츠를 게시할 수 있게 된다.



(그림 1) 기존 트랜스코딩과 분산 트랜스코딩 비교

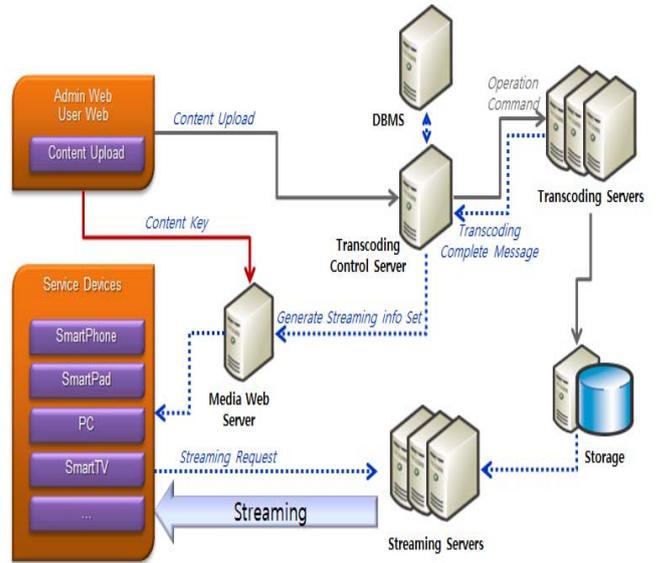
본 연구에서는 상기 그림과 같이 복수의 트랜스코딩 서버를 통하여 웹서비스용 콘텐츠 제공에 필요한 파일로 트랜스코딩하며 현존 브라우저는 H.264 코덱의 .MP4, WebM 코덱의 .WebM을 통하여 콘텐츠 스트리밍 서비스가 가능하므로 이 두 가지의 파일 포맷과 게시에 필요한 Thumbnail Image를 추출해 내는 것을 기본 전제로 하였다.

2.2 시스템 구성

본 연구를 위해 구성한 시스템은 서버 분산 트랜스코딩을 통해 실제 사용자로 하여금 동영상 업로드부터 시청까지 제공하기 위한 트랜스코딩 서버, 콘트롤 서버, 스트리밍 서버, DB 서버 등으로 구성했으며, 구성은 아래 <표 1> 및 (그림 2)과 같다.

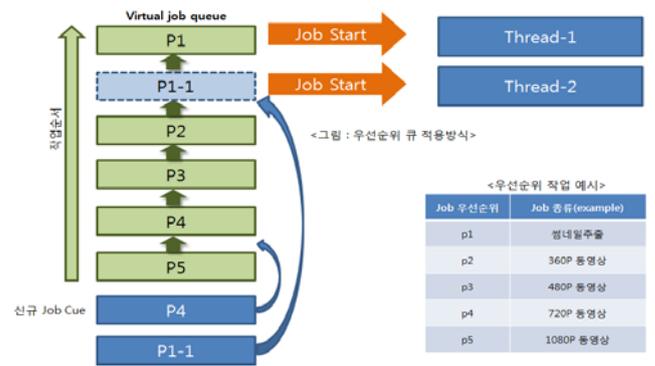
<표 1> 시스템 H/W 종류 및 사양

서버 종류	H/W 규격	수량 (EA)
Transcoding Server	1CPU HexaCore 2.0Ghz, 8GB MEM	12
Control Server	1CPU HexaCore 2.0Ghz, 8GB MEM	2
Web Server	1CPU HexaCore 2.0Ghz, 8GB MEM	3
Streaming Server	1CPU HexaCore 2.0Ghz, 8GB MEM	10
Storage	Usable 30TB, Raid 1/5	1
Switch	1G Switch	1



(그림 2) 시스템 구성도

2.1에서 언급한 바와 같이 본 연구는 분산트랜스코딩을 적용하였으며, 이에 더불어 사용자로 하여금 최단시간에 동영상을 추출하기 위하여, 우선순위 큐 알고리즘을 적용하였다. 우선순위는 썸네일추출(P1), 360P(P2), 480P(P3), 720P(P4), 1080P(P5) 순으로 우선순위를 적용하였다.



(그림 3) 우선순위 큐 알고리즘 적용방식

상기 내용에 대한 처리를 위하여 아래와 같은 우선순위 알고리즘을 코드 내 적용하였다.

```

/* Job의 목록으로부터 우선순위를 정하는 방법 */
var var = from o in db.VW_TranscodingJob_L
          where o.TranscodeOK = false && o.TranscodingServerID = null
          orderby o.Priority descending
          orderby o.ThumbEnable descending
          orderby o.BitrateKB
          select o;
    
```

3. 성능측정 및 결과분석

3.1 성능측정 방법

본 시스템의 성능을 측정하기 위하여 다음과 같은 트

랜스코딩의 규칙에 의해 연구를 수행하고자 한다. ① 트랜스코딩은 FFMPEG 라이브러리 Ver 1.0를 사용한다. ② 콘텐츠 업로드가 완료되면 콘트롤 서버는 각 트랜스코딩 서버에 분산시켜 작업을 할당한다. ③ 트랜스코딩은 9개 작업으로 나누어지며, 그 작업의 결과물은 <표 2>와 같다.

<표 2> 트랜스코딩 파일 종류

Formats	JPG/Image	H.264/MP4	WebM/WebM
Thumbnail Image	O	-	-
360P(200Kbit/s)	-	O	O
480P(500Kbit/s)	-	O	O
720P(1Mbit/s)	-	O	O
720P(2.5Mbit/s)	-	O	O

④ 트랜스코딩 서버는 작업 상태값을 주기적으로 콘트롤 서버에 전달한다. ⑤ 복수 파일의 트랜스코딩은 Bitrate가 낮은 순으로 우선순위를 가지며, Thumbnail 이미지는 모든 동영상 콘텐츠에 최우선적으로 변환한다. ⑥ 각 포맷별, 해상도별 변환이 완료되면 콘트롤 서버는 Callback 주소로 완료된 파일의 종류 및 개수를 통지해준다.

3.2 성능비교 결과

3.2.1 본 시스템을 활용한 비교결과

본 연구에 사용된 파일형식 및 속성은 아래와 같다.

<표 3> 실험파일의 속성

File Name	TEST01.MOV	TEST02.FLV	TEST03.MKV	TEST04.AVI
Format	MPEG-4	Flash Video	MKV	AVI
Running Time	3:01	4:21	3:19	3:09
Size	230MB	316MB	293MB	230MB
Source Size	1280 x 720	1920 x 1080	1920 x 1080	1920 x 1080
FPS	23.99	29.97	23.98	29.97
Bitrate	10.7Mbps	10.2Mbps	12.3Mbps	10.4Mbps
Codec	AVC	FLV	AVC	DIVX

실험에 사용된 4가지의 파일을 트랜스코딩한 결과치는 <표 4>과 같다. 전반적으로 WebM에 비해 MP4 형식의 파일이 트랜스코딩이 빨리 이뤄짐을 알 수 있다. 이는 FFMPEG Library에서 압축 알고리즘상 H.264 CODEC이 WebM CODEC보다 보다 높은 효율성을 제공하기 때문이다.

<표 4> 트랜스코딩 결과

File Name	해상도/비트레이트	MP4	WebM
TEST01.MOV	Thumbnail Source Size	0:00:04	-
	360p, 200Kbits/s	0:00:31	0:01:39
	480p, 500Kbits/s	0:00:47	0:02:58
	720p, 1Mbits/s	0:01:03	0:06:20
TEST02.FLV	Thumbnail Source Size	0:00:08	-
	360p, 200Kbits/s	0:02:37	0:07:45
	480p, 500Kbits/s	0:03:08	0:07:04
	720p, 1Mbits/s	0:02:54	0:11:06
TEST03.MKV	Thumbnail Source Size	0:00:06	-
	360p, 200Kbits/s	0:00:48	0:05:43
	480p, 500Kbits/s	0:01:11	0:05:42
	720p, 1Mbits/s	0:01:19	0:05:39
TEST04.AVI	Thumbnail Source Size	0:00:04	-
	360p, 200Kbits/s	0:00:56	0:04:40
	480p, 500Kbits/s	0:01:17	0:07:39
	720p, 1Mbits/s	0:01:27	0:13:06
TEST04.AVI	720p, 2.5Mbits/s	0:06:27	0:12:31

또한 실험내용을 분석한 결과, 파일별 저해상도부터 고해상도 순으로 트랜스코딩이 완료됨을 나타내었으며 특히, MP4의 경우는 360p(200Kbits/s)는 720p (2.5Mbits/s) 대비 평균 19.53%, WebM의 경우는 58.25% 시간소요를 나타내었다.

3.2.2 YouTube의 트랜스코딩 결과

3.2.1에서 사용한 자료를 활용하여 YouTube를 통해 트랜스코딩 한 결과는 <표 5>과 같다. YouTube의 경우는 모바일 플레이용 3gp 포맷, H.264 CODEC을 지원하지 않는 브라우저를 위한 플래시 플레이어용 FLV 포맷, 그리고 H.264 CODEC의 MP4포맷으로 트랜스코딩하며, 최저 240P로부터 360P, 480P, 720P, 1080P까지 변환하는 트랜스코딩 알고리즘을 가지고 있다.

<표 5> YouTube 트랜스코딩 결과

Format	Transcoding Time(총 완료시간)			
	MOV	FLV	MKV	AVI
3gp (Mobile)	0:20:13	0:26:13	0:26:55	0:28:41
flv (240p)				
flv (360p)				
flv (480p)				
mp4 (360p)				
mp4 (720p)				
mp4 (1080p)				

3.2.3 본 시스템과 YouTube와의 성능비교

아래 <표 6>은 본 트랜스코딩 시스템과 YouTube의 트랜스코딩 시간(3회 평균 시간)과 비교한 표이다.

<표 6> YouTube 트랜스코딩 결과

파일명	본 시스템의 트랜스코딩 시간 (업로드 시간 포함)	YouTube의 트랜스코딩 시간
TEST01.MOV	0:08:00	0:20:13
TEST02.FLV	0:12:50	0:26:13
TEST03.MKV	0:07:22	0:26:55
TEST04.AVI	0:14:30	0:28:41

본 트랜스코딩 시스템과 Youtube 트랜스코딩의 총 완료시간의 평균치를 계산해 봤을 때, 본 시스템은 YouTube에 비해 41.61%로 시간이 단축되었음을 확인할 수 있으며, 이는 보다 진보적인 트랜스코딩 시스템이라 할 수 있겠다.

4. 결론 및 활용방안

본 논문에서는 개선된 트랜스코딩 시스템을 기반으로 보다 신속하게 서비스를 제공을 하기 위한 우선순위 알고리즘 적용한 맞춤형 트랜스코딩 기법을 제안하였다. 향후에도 멀티미디어의 발달에 맞춰 스트리밍 및 트랜스코딩에 관한 다양한 연구들이 활발히 진행될 것이라 전망한다. 또한 본 연구결과는 트랜스코딩 성능개선 뿐만 아니라 모바일 대상자를 위한 맞춤형 서비스를 보다 빨리 제공할 수 있을 것이며, 그 수요는 점차 증대될 것으로 예상된다.

본 연구는 미래창조과학부 및 한국정보화진흥원의 방송통신융합시범사업의 일환으로 수행하였음

참고문헌

- [1] 정유현, 정해원, “멀티미디어 트랜스코딩 기술 동향”, 전자통신동향분석 제19권 제6호 통권90호, pp.83-92, 2004. 12.
- [2] 허난숙, 한우람, 이좌형, 서동만, 김윤, 정인범, “이동사용자를 위한 적응적 트랜스코딩 서비스의 구현”, 한국정보과학회, 2006.
- [3] M. Cho, J. Schlessman, W. Wolf, S. Mukhopadhyay, “Reconfigurable SRAM rchitecture with Spatial Voltage Scaling for Low Power Mobile Multimedia Applications,” Journal IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.19, issue.1, Jan. 2011.
- [4] 최면욱, 이좌형, 방철석, 김병길, 정인범, “트랜스코딩 시스템에서 자원가중치 기반 부하 분산 모델”, 한국정보과학회, 2004.
- [5] <http://www.ffmpeg.org/>