

# 이질 분산환경에서 MPEG 비디오의 병렬 파싱

남윤영<sup>o</sup> 황인준

아주대학교 정보통신전문대학원 정보통신공학과  
{youngman, ehwang}@ajou.ac.kr

## Parallel Parsing of MPEG Video in Heterogeneous Distributed Environment

Yunyoung Nam<sup>o</sup> Eenjun Hwang

Graduate School of Information and Communication, Ajou University

### 요 약

디지털 비디오의 사용 증가로 인해 비디오의 인덱싱, 브라우징, 검색과 같은 서비스가 필요하게 되었다. 이와 같은 서비스의 효율적인 구축을 위한 단계 중, 비디오 파싱은 멀티미디어 애플리케이션에서 중요한 부분을 차지하고 있다. 일반적으로, 비디오 파싱은 복잡한 연산을 하기 때문에, 전통적인 단일 컴퓨터에 의한 파싱 방법은 많은 시간이 소요된다. 이러한 문제는 병렬 및 분산 컴퓨팅을 사용하여 파싱에 필요한 시간을 줄임으로써 해결할 수 있다. 본 논문에서는 이질 분산 환경의 멀티 컴퓨터를 사용한 세 가지 MPEG 비디오의 병렬 스케줄링 알고리즘을 제안하였다. 성능 평가에서는 제안한 스케줄링 알고리즘의 speedup과 load balancing에 관하여 비교 실험하였으며, 제안한 스케줄링 알고리즘을 통해 매우 좋은 성능 향상을 얻을 수 있었다.

### 1. 서 론

지난 몇 년 동안 네트워크의 고속화와 멀티미디어 압축 기술의 발달로 인하여 디지털 미디어의 사용은 급격하게 증가되었으며, 사용자는 더욱더 다양한 검색을 요구하게 되었다. 이에 따라 많은 양의 멀티미디어 데이터를 효율적으로 조직화하고 추출하는 것이 필요하게 되었다.

특히, 디지털 비디오는 일반적인 데이터보다 많은 저장 공간과 네트워크의 대역폭이 필요하다. 이와 같은 요구에 따라 MPEG(Moving Pictures Expert Group)[1]은 디지털 비디오의 압축 표준(MPEG-1, MPEG-2, MPEG-4)을 개발하였다. MPEG은 VOD(Video-On-Demand), MOD(movie-on-demand), DVD(digital video disc), HDTV(high definition television), 전자 도서관과 같은 멀티미디어 애플리케이션에 가장 많이 사용되는 압축 표준이다.

비록, 디지털 비디오가 아날로그 비디오보다 압축과 관리면에서 쉽다는 장점이 있지만, 아직까지는 대부분의 애플리케이션은 다양한 정보를 담고 있는 비디오를 효율적인 조직화와 추출을 하지 못하고 있다. 이러한 문제를 해결하기 위해, 기본적인 텍스트 검색뿐만 아니라 내용 기반 인덱싱이 필요하게 되었다. 즉, 멀티미디어 시스템에서 멀티미디어 데이터들은 메타데이터와 함께 서로 연관하여 결합되어야 한다 [11].

디지털 비디오의 경우, 인덱싱을 위해 샷(shot)과 객체(object)와 같은 요소로 분석하는 연구가 진행되었다. 이러한 비디오 세그멘테이션은 시간적 세그멘테이션과 공간적 세그멘테이션으로 나눌 수 있다. 시간적 세그멘테이션은 샷 경계, 디졸브(dissolve), 페이드(fade), 와이프(wipe)와 같이 일정한 시간내에서 영상의 변화를 검출하는 것을 말한다. 공간적 세그멘테이션은 각 프레임에서 객체의 추출이나 움직임을 파악하는 것을 말한다. 예를 들어 얼굴 검출, 엣지(edge) 검출, 텍스트 검출이 이에 속한다.

이러한 구조화된 비디오가 구조화하지 않은 비디오보다 더 나은 검색을 가능하게 하므로 비디오 파싱은 많은 디지털 비디오 애플리케이션에서 중요한 역할을 차지하고 있다. 그러나 파싱하는데 복잡한 계산이 필요로 하기 때문에, 일반적인 단일 컴퓨터에서의 파싱은 많은 시간

이 소요된다. 특히, DCT(discrete cosine transform), ME(motion estimation), MC(motion compensation)은 복잡한 계산을 하기 때문에 시간적인 비용이 크다. 예를 들어, MPEG-1 비디오(3477 frames, 320×240 pixels, IBPBPBPBP pattern)는 SUN Sparc Ultra-60에서 샷 경계 검출을 하는데 1,086초가 소요된다. 즉, 초당 3.4 프레임의 속도로 샷을 검출한다.

실시간 인덱싱 처리는 멀티미디어 애플리케이션에서 필요하지만, 어떤 경우는 실시간 처리보다 더 빠른 성능을 요구한다. 예를 들어, 전자 도서관에 100만여개의 비디오 타이틀을 분석하고자 할 때, 한 개의 비디오 타이틀을 비디오 파싱하는 데 한시간이 소요된다면, 실시간으로 모두 처리할 경우 100년 보다 더 많은 시간이 소요된다. 하지만, 실시간 처리보다 더 빠르게 처리할 수 있다면 단지 몇 달만에 모두 처리할 수 있다.

본 논문에서는 비디오를 파싱하는 데 필요한 시간을 줄이기 위한 이질 분산환경에서 MPEG 비디오의 병렬 처리를 제안하였으며, 이를 위해 몇 가지 스케줄링 방법을 개발하였다.

본 논문의 구성은 다음과 같다. 2장에서는 MPEG의 개요와 병렬 비디오 파싱에 대해서 알아보고, 3장에서는 병렬 비디오 파싱 모델에 대해 설명한다. 4장에서는 몇 가지 스케줄링 방법에 대해서 설명하고, 5장에서는 실험 결과를 보인다. 마지막으로 6장에서는 결론과 향후 계획에 대해서 논의한다.

### 2. 관련연구

#### 2.1 MPEG 개요

MPEG 비디오는 일련의 GOP(Group of picture)로 구성되며, GOP는 프레임들로 구성된다. 프레임은 프레임에서의 행으로 구성된 부분인 슬라이스(slice)로 나누어지며, 슬라이스는 6개의 매크로블록(macroblock)으로 구성된다. MPEG 비디오는 시간적인 블록 기반의 모션 보정과 공간적인 DCT기반의 압축기법에 의해 압축이 된다. 움직임 정보는 16×16 픽셀 블록을 이용하여 계산되고 공간적 정보화 함께 전달된다. 6개의 블록을 포함하는 매크로 블록은 밝기(Y)를 위한 4개의 블록(8×8 픽셀)과 색채(U와 V)를 위한 2개의 블록(8×8픽셀)으로 구성된다.

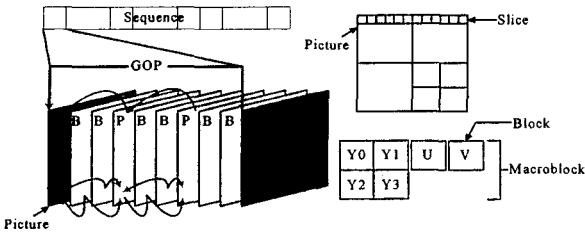


그림 1 I-프레임, P-프레임, and B-프레임으로 구성된 MPEG 비디오의 예

MPEG은 I-프레임 (intra-coded frame), P-프레임 (predictive-coded frame), B-프레임 (bidirectionally-predictive coded frame)으로 나누어 진다. I-프레임은 다른 프레임과 관계없이 인코딩 되고 프레임안에 공간적 상관성을 이용한다. P-프레임은 이전 I-프레임 또는 P-프레임으로부터의 움직임 보상을 이용하여 인코딩한다. B-프레임은 움직임 보상을 위해 이전 또는 이후 프레임 (I-프레임 또는 P-프레임)이 필요하다.

MPEG 비디오의 프레임은 I, P, B-프레임으로 압축이 된다. I, P, B-프레임의 패턴은 비디오 안에서 반복되는 데, 예를 들어 IBBPBPB의 프레임이 하나의 패턴으로 반복된다. 즉, MPEG 비디오는 일련의 GOP로 구성되어 있다. GOP는 적어도 하나의 I-프레임이 포함되어야 하며, 첫 번째 프레임은 I-프레임 또는 B-프레임이고 마지막 프레임은 I-프레임 또는 P-프레임이어야 한다. 따라서 GOP는 독립적으로 디코딩할 수 있다.

2.2 병렬 비디오 파싱

MPEG 비디오의 병렬 파싱은 위에서 설명한 슬라이스, 프레임, GOP 단위로 할 수 있다[3][8][10].

- 슬라이스 단위 : 슬라이스를 작업 큐에 넣은 후 작업 프로세스가 슬라이스를 디코딩하고 DC값의 합을 계산한다. 모든 프로세스는 같은 프레임을 동시에 수행하며 서로 통신을 통해 조정해야 한다.

- 프레임 단위 : 작업 프로세스가 인접한 프레임을 디코딩하며 만약 I-프레임 또는 P-프레임이 없다면 다른 프로세스와 통신을 해야만 한다.

- GOP 단위 : GOP는 독립적으로 디코딩을 할 수 있기 때문에 다른 프로세스와 통신을 할 필요가 없다. 따라서 결과를 더 빠르고 쉽게 얻을 수 있다.

GOP 단위의 병렬 비디오 파싱은 다른 두 가지의 방법보다 통신면에서의 낮은 오버헤드 때문에 속도 향상(speedup) 측면에서 좋은 결과를 얻을 수 있다 [8][10]. 본 논문에서는 좋은 성능을 얻기 위해 GOP 단위로 파싱한다.

3. 병렬 비디오 파싱 모델

병렬 비디오 파싱을 위해 다음과 같은 세 가지 프로세스를 사용한다.

- 스캔 프로세스는 디스크에서 비디오 파일을 읽은 후 프레임 사이즈, GOP수, 평균 비트율같은 정보를 수집후 작업 큐에 작업단위로 넣는 역할을 한다. 스캔의 속도는 압축형태와 파일 사이즈에 따라 달라지며 분석된 비디오는 작업 프로세스로 옮겨진다.

- 작업 프로세스는 GOP의 단위로 계산을 한다. 시스템은 여러 개의 슬레이브로 구성되고 작업 요청과 결과를 보내기 위해 마스터와 통신을 한다.

- 결합 프로세스는 슬레이브에게 받은 결과를 입력형태의 순서로 재정렬하는 역할을 한다.

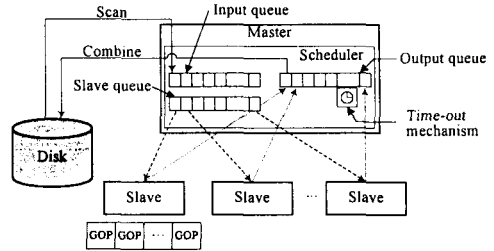


그림 2 시스템의 구조

그림 2는 전형적인 마스터-슬레이브 모델인 시스템의 구조를 나타낸다.

4. 스케줄링 알고리즘

효율적인 병렬 파싱을 위해 라운드 로빈(Round Robin), 사이즈 적응적인 라운드 로빈(Size-Adaptive Round Robin), 동적으로 사이즈 적응적인 라운드 로빈(Dynamic Size-Adaptive Round Robin) 이라는 세 가지 스케줄링 알고리즘을 사용하였다.

4.1 라운드 로빈(Round Robin)

라운드 로빈 스케줄링은 그림 3과 같이 입력 큐에 있는 데이터를 슬레이브들이 나누어 분석하는 방식이다. 예를 들어, 그림 3처럼 마스터(M)와 세 개의 슬레이브(S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>)가 있다면, 우선 큐에 있는 순서대로 S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>에게 작업을 분배한다. S<sub>3</sub>이 시간 T+(i>0)에 먼저 끝났기 때문에 M과 먼저 통신을 하고, S<sub>2</sub>와 S<sub>1</sub>는 순서대로 슬레이브 큐에 저장되며 S<sub>3</sub>가 M과 통신이 끝나면 슬레이브 큐에 있는 S<sub>2</sub>와 S<sub>1</sub>가 차례대로 M과 통신을 한다.

이 방식은 하나의 슬레이브가 다른 슬레이브보다 작업 처리 시간 또는 마스터와 통신하는 시간이 많아지면 마스터 또는 다른 슬레이브들이 휴지(idle)상태로 남아있게 된다. 이러한 문제를 해결하기 위해서 개선된 라운드 로빈 스케줄링이 필요하다.

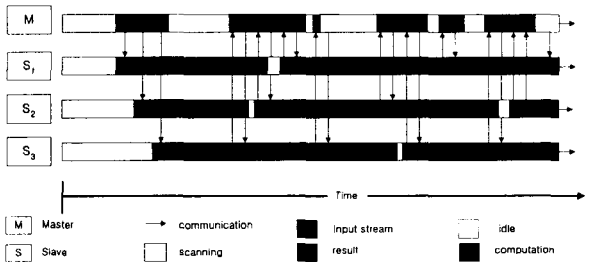


그림 3 라운드 로빈 스케줄링

4.2 사이즈 적응적인 라운드 로빈(Size-Adaptive Round Robin)

슬레이브는 시스템의 성능에 따라 계산시간이 서로 다르며, 네트워크 환경의 전송 속도는 항상 변하기 때문에 슬레이브들은 서로 다른 네트워크 대역폭을 가지게 된다. 이러한 이유로 각 슬레이브의 통신시간과 계산시간은 효율적인 스케줄링을 위해서 고려되어야 한다.

그림 4는 사이즈 적응적인 라운드 로빈 스케줄링을 보여주고 있다. 마스터는 작은 샘플파일을 각 슬레이브들에게 보내서 슬레이브들의 계산성능과 네트워크 대역폭을 알아낸다. 이를 이용하여 마스터는 각 슬레이브들에게 전송할 비디오 스트림의 사이즈를 조정한다. 이렇게 조정된 비디오 스트림은 높은 성능을 지닌 슬레이브 순서로 분배된다. 그림 4에서는 성능이 높은 순서가

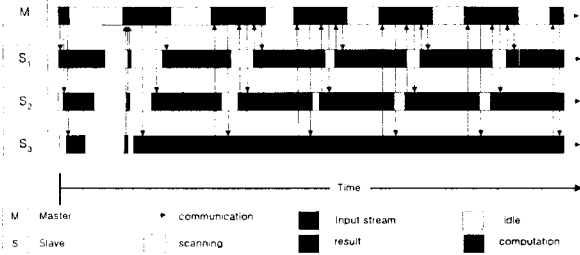


그림 4 사이즈 적응적인 라운드 로빈

$S_3 > S_2 > S_1$ 이기 때문에  $S_3, S_2, S_1$  순서로 분배된다. 그러나, 마스터와 슬레이브가 데이터 부족으로 인해 휴지상태에 놓일 수 있다. 마스터의 경우는 로드 밸런싱(load balancing)을 통해 조절할 수 있으며, 슬레이브 경우는 대기하는 동안 다른 작업을 수행하는 방법이 있다. 본 논문에서는 이를 위해 동적으로 사이즈 적응적인 라운드 로빈 스케줄링을 제안하였다.

4.3 동적으로 사이즈 적응적인 라운드 로빈(Size-Adaptive Round Robin)

위와 같은 문제를 해결하기 위해 논 블로킹(non-blocking) 전략이 고려되어야만 한다. 마스터의 경우, 슬레이브의 수를 충분히 증가시켜 마스터의 휴지 상태를 해결할 수 있지만, 이 경우 많은 수의 슬레이브가 휴지 상태가 되어 자원의 낭비를 초래하게 된다. 슬레이브의 경우, 슬레이브의 계산성과 대역폭을 예상하여 작업의 크기와 슬레이브의 수를 계속 조정함으로써 휴지시간을 줄일 수 있으며 자원을 효율적으로 사용할 수 있다.

5. 실험결과

실험을 위하여 SUN Sparc Ultra-60과 이더넷(ethernet)과 연결된 10개의 슬레이브 머신을 사용하였다. 슬레이브들은 서로 다른 계산 성능(0.8GHz ~ 1.8GHz)과 네트워크 처리량(network throughput, 10Mbps ~ 100Mbps)을 가지고 있다. 즉, 전형적인 이질 분산 멀티 컴퓨터로 구성되었다. 제안한 스케줄링의 성능을 측정하기 위해서 세 가지 서로 다른 비디오 데이터를 가지고 실험하였으며, 비디오는 MPEG-1, MPEG-4로 압축되어 있다. 일반적으로 멀티컴퓨터 환경은 데이터 표현과 메시지 교환 프로토콜을 통해 이루어지는데, 본 시스템에서는 XML(Extensible Markup Language)[2]과 SOAP(Simple Object Access Protocol)[2]을 사용하였다. 그림 5는 슬레이브 수에 따른 speedup의 변화를 보여주고, 그림 6은 파일 크기가 700MB인 영화에 라운드 로빈(RR) 스케줄링과 사이즈 적응적인 라운드 로빈(SARR) 스케줄링을 사용하였을 때 슬레이브 수에 따른 speedup의 변화를 보여주고 있다.

그림 5에서 보듯이, 뉴스의 경우 8개의 슬레이브를 사용하였을 때 단일 컴퓨터보다 7.4배의 빠른 성능을 보였으며, 그림 6에서는 SARR이 RR보다 더 좋은 성능을 보이고 있다. 그림 6에서는 11개의 슬레이브를 사용하였을 때, SARR이 RR보다 9.5%, 단일 컴퓨터보다 7.6배의 성능 향상을 보였다. 그림 7은 동적으로 사이즈 적응적인 라운드 로빈 스케줄링을 사용했을 때, 시간에 따른 슬레이브 수의 변화를 보여주고 있다.

6. 결과

본 논문에서는 이질 분산 환경에서 MPEG 비디오의 병렬 파싱을 제안하였다. 속도 향상을 위해 GOP 단위로 파싱하였으며, 세 가지 스케줄링 알고리즘을 제안하고 성능을 실험하였다.

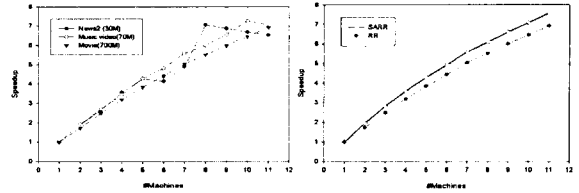


그림 5 Speedup vs. #Machine

그림 6 Speedup vs. #Machine

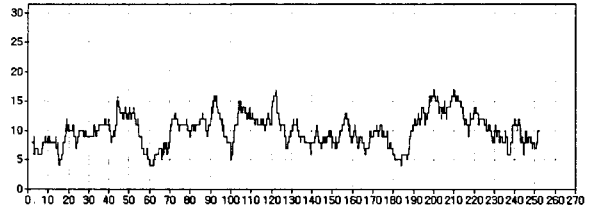


그림 7 #Machine vs. 시간

슬레이브의 수가 고정되어 있을 때는 사이즈 적응적인 라운드 로빈 스케줄링이 좋은 성능을 보였으며, 11개의 슬레이브를 사용하였을 때, 단일 컴퓨터보다 7.6배의 속도 향상을 보였다. 동적으로 사이즈 적응적인 라운드 로빈 스케줄링은 네트워크 환경이 신뢰할 수 없고 변화가 심한 분산 환경에서 적합한 스케줄링 방법이다.

본 논문에서 제안한 MPEG 비디오의 병렬 파싱은 이질 분산환경에서 이식 가능하고 유연하게 사용될 수 있다. 향후 계획은 좀더 효율적이고 최적화된 병렬 스케줄링 방법을 개발하는 것이다.

참고문헌

- [1] "The MPEG home page", <http://mpeg.telecomitalia.com/>
- [2] "World Wide Web Consortium (W3C) home page", <http://www.w3.org/>
- [3] Kevin L. Gong, et al, "Parallel MPEG-1 Video Encoding." The Picture Coding Symposium, 1994.
- [4] Jeffrey Moore, et al, "Optimal parallel MPEG encoding", Cornell University, 1994.
- [5] Xian-He Sun, Rover, D.T. "Scalability of parallel algorithm-machine combinations", Parallel and Distributed Systems, IEEE Transactions, Vol. 5(6), pp. 599-613, 1994
- [6] K. Shen, et al, "A Parallel Implementation of an MPEG1 Encoder: Faster than Real-Time," The SPIE Conference on Digital Video Compression, 1995.
- [7] Foster, Designing and Building Parallel Programs, Addison-Wesley, 1995.
- [8] A. Bilas, et al, "Real time parallel MPEG-2 decoding in software". The 11th International Parallel Processing Symposium, 1997.
- [9] V. Kobla, et al, "Indexing and Retrieval of MPEG Compressed Video", Journal of Electronic Imaging, Vol. 7(2), pp. 294-307, April, 1998.
- [10] Suchendra M. Bhandarkar, et al, "Parallel Parsing of MPEG Video", International Conference on Parallel Processing (ICPP), 2001.
- [11] K.R. Rao, et al, Multimedia Communication Systems, Prentice Hall, 2002.