적응형 멀티미디어 시스템에서 Layered Stream의 배치

이흥기°*, 김현정*, 김상형*, 이성인*, 김두현**, 유관종*
충남대 컴퓨터 과학과 * , 전자 통신 연구소 **

{helius, hjkim, shkim, ppuyo}@cnu.ac.kr, silee@health.ac.kr, doohyun@etri.re.kr, kjyoo@cnu.ac.kr

Placement layered streams in adaptive multimedia system

Heung-ki Lee° * , Hyun-Jeong Kim * , Sang-Hyoung Kim * ,
Sung-In Lee * , Doo-Hyun Kim ** , Kawn-Jong Yoo *

Department of computer science, Chung-nam University *,
Electronics and Telecommunications Research Institute **

요 약

변화하는 네트웍환경에서, VoD서비스를 제공하는 방법에 대해서 많은 연구가 이루어지고 있다. 이들 중에서 스트림 데이터의 특성을 이용하여, 데이터를 제공하는 방법에 대해 활발한 연구가 진행중이다. 그러나, 이 연구들은 전송에만 초점을 많이 맞추고 있다. 따라서, 본 연구는 이러한 스트림의 저장 방법에 대해서 연구한다.

1. 서 론

네트웍 환경의 발전과 개인 컴퓨터 성능의 발전은 우리의 생활에 많은 변화를 주었다. 많은 변화들 중에 두드러진 것이 인터넷을 이용한 서비스라고 할 수 있다. 우리는 이러한 서비스를 이용하여 여러 가지 편리한 서비스들을 사용하고 있다.

이런 인터넷을 이용한 서비스들 중에서 가장 주목받고 있는 것 중에 하나가 VOD(Video-On-Demand)서비스이다. 그러나, VOD 서비스에는 아직 여러 문제점들이 산재해 있다. QoS보장 문제와 병렬 VOD서비스에서 서버측의 load를 분산시키는 방법에 관한 것들이다. 이러한 문제점에 대하여 알아보도록 하겠다.

본 논문은 2장에서는 기존에 연구되어진 MPEG 멀티미디어 데이터를 scalable하게 나누는 방법과 동적으로 전송 데이터 양을 변화시키는 방법에 대해서 살펴본다. 그리고 3장에서는 이렇게 나누어진 멀티미디어 데이터를 서버에 저장하는 방법에 대해서 알아볼 것이다. 그리고 4장에서는 결론을, 5장에서는 향후 발전 과제를 기술한다.

2. Scalability

Scalability는 기존의 MPEG 시스템에 이미 존재하는 기능이다. 그러나, 나누어진 양의 크기가 여전히 크다는 문제점을 나타내고 있었다. 이에 많은 연구를 통해서, frame을 시간별로 나누는 temporal방식 과 공간별로 나누는 spatial방식을 동시에 적용하게 되었다. 그 결과로 본래의 stream에 비해서 10 % 미만의 크기를 가지는 base layer를 만들 수 있었다.

2.1 Temporal Scalability

프레임마다 존재하는 의존성을 이용하여 stream을 나누는 temporal scalability가 있다. 즉, 독립적인 성격을 가지고 있는 I-Picture를 base layer로 분류하며, 이 base layer를 참조하는 P-Picture와 B-Picture를 각각 1-level enhancement layer, 2-level enhancement layer로 나누는 것이다. [1]

2.2 Spatial Scalability

다른 방법은 spatial scalability를 적용하는 방식이다. 이것은 block의 정보가 저주파를 나타내는 부분과 고주파를 나타내는 부분으로 나누어지는 것을 이용하는 것이다. 이것은 화면 전체의 평균값을 나타내는 저주파 영역과 변화정도를 나타내는 고주파 영역을 나눔으로써, 화면의 변화를 나타내는 것이다. 총 5개의 layer로 나누어진다. [1]

위에서 기술한 두 가지 방법을 하나의 stream에 동시 적용을 수행함으로써, 총 15개의 layer를 생성할 수가 있다. 이러한 방식으로 생성된 stream은 네트웍의 상황에

따라서, 동적으로 전송 여부를 결정한다. [1]

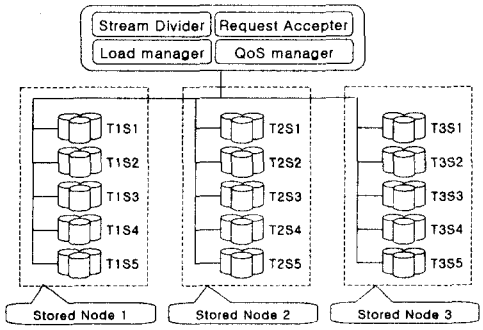
3. Load Balance

3.1 Placement of Layered stream

동적으로 전송되어지는 데이터의 양을 조절하기 위해서, 나뉘어진 stream의 임의의 위치를 접근하여야 한다. 시스템의 성능을 위해서, 나뉘어진 stream단위로 데이터 저장을 수행한다.

본 연구에서 stream은 temporal과 spatial을 기준으로 나뉘어 졌다. 이러한 기준은 저장방식에서도 적용되어진다. 즉, temporal 데이터를 기준으로 삼아 stream을 분산시키는 방법과 다른 하나는 spatial 데이터를 기준으로 삼아 stream을 분산을 시키는 방법이다.

Temporal정보가 기준인 저장방식은 picture 정보에 대해서 분산시킴으로써, load balancing을 가져올 수 있다. 즉, temporal 데이터를 중심으로 배치한 후, spatial 데이터를 중심으로 배치한다. 그러나 이러한 방식은 단위시간당 필요로 하는 데이터가 하나의 노드에 집중되어지는 현상이 발생되어질 수 있다. 즉, 하나의 frame을 표현하기 위해서, 필요한 데이터가 하나의 노드에 저장되어져 있기 때문에 이러한 현상이 발생할 수 있다.

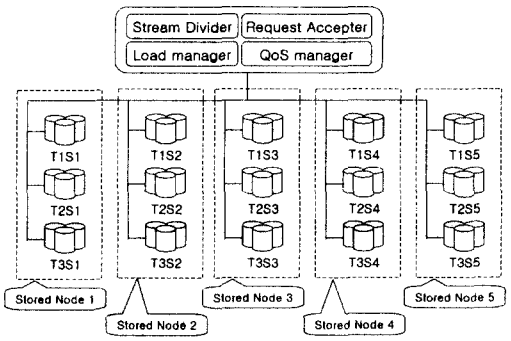


[그림 1] Striping by temporal layering

위의 그림에서 control 부분은 stored node들을 관리하여주는 역할을 수행한다. "Stream Divider"는 stream을 나누는 일을, "Request Accepter"는 서비스 요청을 받아들이는 작업을, "Load manager"는 stroed node들의 관리를, 마지막으로 "QoS manager"는 전송 중의 Network상태에 대한 점검을 수행한다.

Spatial정보가 기준인 방식은 spatial 정보를 중심으로 배치를 수행한 후에, 다시 temporal 데이터를 중심으로 데이터를 배치시키는 방식이다. 이것은 spatial 정보별로 stream이 나뉘어져 저장되어짐으로써, load balancing 효과를 가져올 수 있다. 이 방식은 하나의 frame에 대한 정보를 하나의 노드에 집중시키지 않고 여러 노드에 분산시켜준다. 그러나, 데이터를 작은 크기로 읽어들임으로써, 효율을 저하시키는 결과를 초래할 수 있다.

Control 부분의 역할은 Temporal 데이터를 기준으로 나뉘어진 방식과 같은 역할을 담당한다.



[그림 2] Striping by spatial layering

3.2 Redundancy of stream data.

보통의 병렬 VoD는 데이터를 중복 저장한다. 이런 중복된 데이터는 하나의 Node가 동작하지 못하는 상황에서 사용되어진다. 그러나, 이러한 중복 저장 방식은 전체 데이터를 중복 저장하거나, 전체 데이터의 연산 결과를 저장하는 방식을 택하고 있다. 그러나 layering 되어진 stream은 stream간의 의존성을 이용하여 중요도를 결정할 수 있다. 우선, stream 간의 의존성을 정리하면 다음과 같다.

1. stream을 보여주기 위해서는 base layer인 T_1S_1레벨이 필요하다.
2. $T_n(1 < n \leq 3)$레벨의 데이터를 표현하여주기 위해서는 적어도 $T_{n-1}S_1$레벨의 데이터가 필요하다.
3. $T_nS_m(1 \leq n \leq 3,\ 1 < m \leq 5)$ 레벨의 데이터를 표현하기 위해서는 T_nS_1레벨의 데이터가 필요하다.
4. $T_nS_m(1 \leq n \leq 3,\ 1 < m \leq 5)$ 레벨을 나타내기 위해서는 T_nS_{m-1}이 필요하다.

이러한 stream의 의존성을 이용하여 Priority를 적용하면, 다음과 같은 표을 얻을 수 있다.

[표 1] Layered Stream Priority

① T_1S_1	② T_1S_2	③ T_1S_3	④ T_1S_4	⑤ T_1S_5
② T_2S_1	③ T_2S_2	④ T_2S_3	⑤ T_2S_4	⑥ T_2S_5
③ T_3S_1	④ T_3S_2	⑤ T_3S_3	⑥ T_3S_4	⑦ T_3S_5
①② ... : Dependency T_nS_m : Layered Stream				

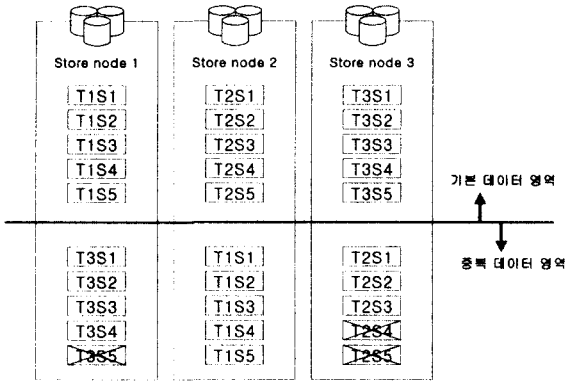
하나의 스트림이 삭제되어질 경우, 삭제되어진 스트림의 하위 스트림도 같이 삭제되어져야한다. 중복 저장된 데이터의 삭제 할 경우에, 다음과 같은 규칙을 따른다.

1. T_1S_1레벨의 데이터가 삭제되어질 경우, 모든 중복 데이터를 삭제한다.
2. $T_nS_1(1 < n \leq 3)$레벨의 데이터가 삭제되어질 경우, 하위 레벨과 같은 레벨의 temporal 데이터가 삭제되

어져야 한다.

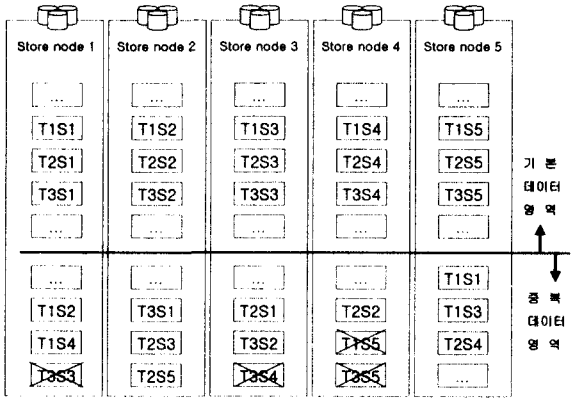
3. $T_nS_m(1 \le n \le 3,\ 1 < m < 5)$ 레벨의 데이터가 삭제되어질 경우, T_nS_m에서 T_nS_5의 데이터가 삭제되어져야 한다.

데이터를 중복 저장하는 방법에 대해서 알아보도록 하겠다. Temporal 정보를 우선 순위로 striping을 수행한 경우에는 'store node 1'에 저장된 데이터를 'store node 2'에 중복저장하고, 'store node 2'에 저장된 데이터를 'store node 3'에 중복 저장하는 형태이다. 이 경우에, 공간이 부족할 경우에 각 노드의 중복 저장되어진 데이터 중에서 가장 하위 레벨의 데이터를 삭제한다.



[그림 3] Redundancy placement in Temporal Striping

Spatial 정보를 우선 순위로 striping을 수행한 경우에는 위의 방식의 적용이 어렵다. 만일 위의 방식을 적용할 경우, Priority가 높은 데이터의 중복 데이터가 임의의 노드에 집중되어지게 된다. 이러한 집중현상을 없애기 위해, 다음과 같은 형태로 striping을 수행한다.



[그림 4] Redundancy data in Spatial Striping

즉, 4번 노드에서 시작하여, Priority의 역순으로 중복 저장을 수행하는 방식이다. 이때, T_2S_1레벨은 본래 stream이 저장되어져 있는 node에 중복 저장된다. 같은 Priority를 가지는 T_1S_2와 T_2S_1이 중복 저장 위치를 바꾼다.

4. 결론

가변적으로 변화하는 인터넷 환경에서, 변화되어지는 양에 능동적으로 대처하는 실험이 꾸준히 연구되어져 왔다. 이러한 기존의 연구는 변화하는 네트웍환경에서 좋은 성능을 보여주고 있다. 그러나 기존의 연구가 네트웍의 전송에만 초점을 맞추어져서 진행되어졌다.

이에 본 논문은 효율적인 저장 방법에 대해서 기술하였다. load를 분산시키는 방법과 중복데이터의 양을 조절하는 방법에 대해서 알아보았다. 이러한 방법을 통해서, layering 되어진 stream을 효율적으로 저장할 수 있다.

5. 향후 계획

현재까지의 시스템은 주로 MPEG-1,2를 기반으로 연구되어져 왔다. 이러한 시스템을 MPEG-4로 확장하는 방안에 대해서 연구가 이루어질 것이다. MPEG-4를 적용하면서, 새로운 형태의 layering개념의 도입과 현재의 layering을 강화시키는 방안에 대해서 연구가 이루어질 것이다. 또한, load balance효과를 방안에 대해서 연구가 이루어질 것이다.

6. 참고 문헌

[1] 김현정, 이홍기, 손호신, 유우종, 김두현, 유관종, "동적 QoS 적응을 위한 Temporal-Fidelity Scaling 기법에 관한 연구," 한국정보처리학회 추계 학술대회, 제7권 2호, pp. 691-694

[2] 김태영, 손호신, 유우종, 김형철, 유관종, "QoS를 고려한 SpatioTemporal Layered Coding and Scalable Transmission에 관한 연구," 한국정보과학회 추계 학술발표회, 제26권 2호(III), pp.614-616, 1999.10

[3] International Standard ISO/IEC 14496-2, Information technology coding of audio-visual object-Part 2 : Visual, 1999

[4] ISO/IEC JTC1/SC29/WG11 N3312, MPEG-4 Video Verification Model version 16.0, March 2000

[5] Jack Y.B. Lee, "Parallel Video SErvers: A Tutorial, " IEEE Multimedia, 1998