

주문형 비디오 서버의 성능 개선을 위한 논리적 디스크 분할 기법

김운석*○ 김지홍** 민상렬*

* 서울대학교 컴퓨터공학과

** 서울대학교 전산학과

Logically Partitioned Disk Arrays for Video-On-Demand Servers

Woonseok Kim* Jihong Kim** Sang Lyul Min*

* Dept. of Computer Engineering, Seoul National University

** Dept. of Computer Science, Seoul National University

요 의

프로세서와 네트워크의 성능이 향상됨에 따라 이들을 이용한 많은 멀티미디어 응용물들이 개발되었다. 특히 인터넷의 발전은 이러한 멀티미디어 응용물들로 하여금 여러 곳에 분산되어 있는 데이터들을 쉽게 이용할 수 있도록 하였다. 멀티미디어 응용물들이 요구하는 데이터는 그 크기가 크고, 실시간 전송을 요구한다는 특성을 가진다. 따라서, 이러한 데이터를 제공하는 서버는 여러 데이터를 다수의 사용자에게 일정한 전송률로 제공할 수 있어야 한다. 본 논문에서는 디스크 배열을 이용하는 주문형 비디오 서버에 있어서 디스크 탐색 시간으로 낭비되는 디스크 대역폭을 최소화하는 논리적 디스크 분할 기법과 이에 적합한 비디오 파일 배치 기법을 제시한다. 추가적으로 얻어질 디스크 대역폭들은 보다 많은 사용자 요구들을 처리하기 위해 사용되거나 비동기적인 사용자 요구들을 처리하는데 이용될 수 있는데, 이를 스트림 지원에 이용할 경우, 평균 3~11% 정도의 스트림수 증가가 있음을 모의 실험을 통해 확인하였다.

1. 서 론

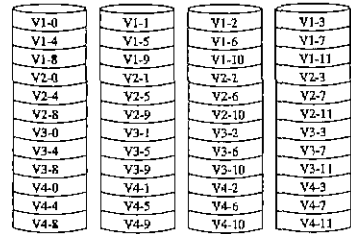
네트워크 기술의 발전과 디지털 압축 기술의 발전은 서버로부터 원거리에 위치한 사용자들에 대해 멀티미디어 서비스의 제공을 가능하게 하였다. 특히, 광섬유와 같은 매체를 이용할 경우, 초당 수 기가바이트의 데이터 전송이 가능하며, 표준화된 동영상 압축 기법들(예: MPEG-1)을 사용할 경우 상대적으로 적은 대역폭으로도 동영상 정보를 지원할 수 있다. 이러한 기술 발전은 네트워크를 통하여 멀티미디어 데이터를 전송받아 서비스 할 수 있는 많은 응용 프로그램 개발들의 확대를 가져왔다. 화상회의, 원격 강의, 그리고 주문형 비디오 및 오디오가 그 대표적인 예라 할 수 있겠다. 비디오 데이터는 다른 데이터들에 비해 상대적으로 크기가 크며, 지속적인 주기적인 데이터 전송을 필요로 한다는 특성을 가진다 [1]. 이러한 대규모의 데이터를 다량으로 저장하기 위해서는 디스크 배열과 같은 저장장치를 사용하여야 하고, 주기적 데이터 전송을 위해선 디스크 대역폭과 같은 자원을 적절히 사용자 요구들에 할당하여야 한다.

본 논문에서는 디스크 배열을 이용하는 주문형 비디오 서버에 있어서, 디스크 탐색 구간을 최소화하여 동시 지원 가능한 사용자의 수를 늘릴 수 있는 방법을 제시하고, 이를 효율적으로 이용하기 위한 논리적 디스크 분할 기법과 비디오 오디오 파일 배치 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 멀티미디어 서버에 대한 연구들에 대해 알아보고, 3장에서는 본 논문에서 기반하고 있는 서버 모델을 소개한다. 4장과 5장에서는 각각 디스크 탐색 시간을 최소화하기 위한 논리적 디스크 분할 기법과, 이에 적합한 비디오 파일 배치 기법을 설명한다. 6장에서는 기존에 제안된 기법들과 본 논문에서 제안하는 기법들에 대한 모의 실험 결과를 보이고, 끝으로 7장에서 결론을 내린다.

2. 관련 연구

멀티미디어 데이터를 제공하는 서버는 주어진 시스템 환경내에서 가능한 많은 사용자들을 서비스 하기 위해 시스템 자원들을 최대한 활용하여야 한다. 특히, 디스크 배열을 이용하는 서버들의 경우, 디스크 대역폭과 메모리 같은 자원의 활용도는 단일 디스크 참조에 의해 읽어들이는 데이터 블록의 크기와 디스크 스케줄링 방법들에 의해 결정되므로 신중히 설정되어야 한다. 기존의 연구들중 [2]에서 Shenoy는 디스크들 간의 작업 부하의 차이가 시스템이 지원할 수 있는 스트림수에 영향을 주게 됨을 지적하고, 적절한 블록 크기를 설정하여 해결할 수 있는 방안을 제시하였다. 이에 반해 Ozden은 [3]에서, 블록 크기에 의해 시스템의 설계 비용과 지원 가능한 스트림의 수가 결정됨에 착안하여, 최적의 설계 비용으로 최다의 스트림을 지원할 수 있는 최적 블록 크기를 구하는 분석적인 모델을 제시하였다. 이들은 블록 크기가 어느 정도까지 증가하면 지원 가능한 스트림의 수는 증가하지만, 이에 따라 요구되는 메모리의 양도 증가하여 시스템 설계 비용이 증가됨을 설명하고, 이에 대해 일



디스크 0 디스크 1 디스크 2 디스크 3

그림 1: Round-robin 방식으로 저장된 데이터

정 스트림 수를 지원하기 위한 최적 비용의 시스템 설계 방안을 제시하였다. 하지만, 이에 사용된 서버 모델은 디스크 헤드 이동 구간에 대해 최악의 경우를 가정했기 때문에 실제 이용되는 디스크 대역폭과는 상당한 차이가 있다.

3. 디스크 기반 서버 모델

본 절에서는 [3]에서 사용된 디스크 기반 서버 모델에 대해 살펴본 후, 이에 개선되어야 할 점을 알아보도록 한다.

3.1 사용자 요구 처리

주문형 비디오 서버에서 사용자 요구와 데이터 전송은 모두 네트워크를 통해 이루어진다. 네트워크를 통해 요청된 사용자 요구들은 일단 사용자 요구 큐에 저장된 후 서비스 시작을 기다리게 되는데, 서버는 현재 남아있는 디스크 대역폭과 서비스 제공에 필요한 다른 시스템 자원들(예: 메모리)의 현재 상태를 고려하여 새로운 사용자 요구에 대한 서비스 제공 여부를 결정짓는다.

서비스 제공이 시작된 사용자 요구에 대해서는 주기적으로 데이터를 전송하여야 하는데, 그 전송률은 클라이언트 측에서 데이터 부족으로 인한 스트림 단절이 이루어 지지 않도록 충분히 높아야 한다. 따라서, 서버는 이와 같은 조건을 만족시키는 한도내에서 일정 시간 간격으로 데이터를 디스크로부터 읽어들이어야 한다. 이때 이 시간 간격을 주기라 한다. 매 주기마다 같은 디스크로부터 데이터를 읽어들이게 되는 스트림들은 하나의 서비스 그룹을 형성하게 되는데, 서비스 그룹의 수는 디스크의 수와 지원되는 미디어 형식의 데이터 소모율에 따라 결정된다. 본 논문에서는 데이터 소모율이 디스크 대역폭 보다 작은 것을 가정하며, 서비스 그룹의 수는 디스크의 수와 일치하는 것으로 한다.

표 1 사용된 기호들과 그 의미

d	블록 크기
q	단일 디스크로 부터 동시 지원되는 스트림의 수
r_{disk}	비디오 파일의 데이터 소모율
t_{seek}	최대 디스크 탐색 시간
t_{rot}	최대 디스크 회전 지연 시간
t_{settle}	디스크 헤드 정착 지연 시간
r_{disk}	최소 디스크 전송률

비디오 파일들이 그림 1와 같이 라운드-로빈(Round-robin) 형태로 저장되어 있고, 각 서비스 그룹들이 이번 주기에 읽어들여야 하는 블록의 번호가 다음과 같고 하자

- 서비스 그룹-0 : V1-0, V2-0, V3-0, V4-0
- 서비스 그룹-1 : V1-3, V2-3, V3-3, V4-3
- 서비스 그룹-2 : V1-6, V2-6, V3-6, V4-6
- 서비스 그룹-3 : V1-9, V2-9, V3-9, V4-9

서비스 그룹-0은 이번 주기에 디스크 0을 이용하고, 다음 주기에는 디스크 1로 부터 잇따르는 데이터를 읽어들이는 것이다. 이때, 다른 서비스 그룹들은 각각 잡어 디스크들을 이용할 것이다. 각 디스크들이 CSCAN 디스크 스케줄링 [4]을 한다고 할때, 서비스 그룹-3은 이전 주기에 디스크 0으로 부터 V4-8 블록을 마지막으로 읽어들었을 것이다. 따라서, 서비스 그룹-0이 디스크 0을 이용하기 위해서는 디스크 헤드를 우선 V4-8의 위치에서 V1-0의 위치로 이동 시키야 한다. 이때 이동되는 구간은 최대 디스크 탐색 구간과 비슷하며, 모든 서비스 그룹들은 매 주기마다 이러한 초기 디스크 탐색 시간을 소요하여야 한다.

3.2 서비스 제어

위와 같이 스트림들을 서비스 할 경우, 한 주기내에 한 서비스 그룹이 수용할 수 있는 스트림의 수는 [3]에서처럼 다음과 같이 나타낼 수 있으며, 이에 사용된 기호들의 의미는 표 1과 같다.

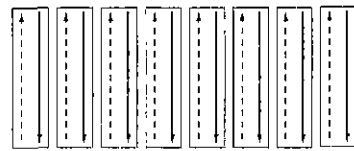
$$q \left(\frac{d}{r_{disk}} + t_{rot} + t_{settle} \right) + 2t_{seek} \leq \frac{d}{r_{disk}} \quad (1)$$

여기서, 한 주기의 길이를 $\frac{d}{r_{disk}}$ 로 나타낼 수 있는데 이 d 크기 만큼의 블록을 읽어 들였을 때 이를 클라이인트 측에서 소모하는데 걸리는 시간을 의미한다. 즉, 한 블록을 전송후에 다음 블록을 준비해 두어야 하는 때까지의 시간 간격을 말한다. 여기서 최대 탐색 시간이 두 번 소요되는데 이는 이전 서비스 그룹에 의해 이동된 디스크 헤드의 위치를 시작점으로 이동 시키기 위한 디스크 탐색 시간과 실제 데이터를 읽기 위해 이동하는데 소요되는 디스크 탐색 시간들의 합을 의미하는 것이다. 식(1)에서 알 수 있듯이 블록 크기를 일대일로 하는냐에 따라 주기의 길이와 한 서비스 그룹이 수용할 수 있는 스트림의 수가 결정된다. 즉, 블록 크기 d를 증가시키면 주기가 길어져 한 주기 내에 한 서비스 그룹이 수용할 수 있는 스트림의 수도 늘어나게 된다. 여기서 주의해야 할 점은, 블록 크기와 스트림 수가 증가하면 요구되는 메모리의 양도 더불어 증가한다는 것이다. 디스크로부터 읽혀진 데이터들을 네트워크를 통해 클라이인트 측에 전달하기 위해서는 일단 메모리에 저장하여야 하고 클라이인트 측이 이를 다 소모하기 이전에 다음 데이터를 준비해 두어야 하므로, 각 스트림에 필요한 메모리의 크기는 2d가 된다. 블록 크기를 증가시켜 지원할 수 있는 스트림의 수를 늘려주며 원 경우 요구되는 메모리의 증가폭도 커져 가격적인 면에서 매우 불리해진다. 블록의 크기는 시스템 설계시 결정되므로, 위와 같은 방법으로 수용할 수 있는 스트림의 수를 미리 결정할 수도 있다. 하지만, 식(1)은 디스크 검색시 최악의 경우를 가정한 것이므로 실제 q개의 스트림을 처리함에 있어 $\frac{d}{r_{disk}}$ 보다 적은 시간이 소요된다. 이때, 최악 디스크 탐색 시간의 발생을 항상 억제하여 남게되는 대역폭의 양이 하나의 블록을 읽어들이기에 충분하다면, 추가적인 스트림 제공이 가능해진다.

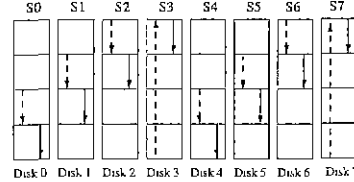
4. 논리적 디스크 분할 기법

본 절에서는 스트림들을 서비스 그룹에 할당함에 있어 한 그룹이 낭비하게 되는 디스크 대역폭을 최소화 할 수 있도록 디스크들을 논리적으로 분할하여 사용하는 방법을 제시한다.

이전 절의 예에서 단일 서비스 그룹-3이 비디오-1에 대한 스트림들만을 포함하였고, 나머지 그룹들도 특정 비디오에 대한 스트림들만을 포함하였다면, 데이터 블록들을 읽어들이기 위해 소요되는 디스크 탐색 시간은 줄어들게 된다. 또한, 만일 이전 서비스 그룹에 의해 마지막으로 참조된 블록의 위치가 현재의 서비스 그룹이 처음으로 참조하고자 하는 블록의 위치와 인접하다면, 초기 디스크 탐색 시간도 줄어들게 된다. 논리적 디스크 분할 기법은 디스크 배열을 논리적으로 분할하여 서비스 그룹들이 특정 영역만을 전담케 함으로써 디스크 탐색으로 인해 낭비되는 디스크 대역폭을 최소화하고 이를 이용해 보다 많은 스트림들을 지원하거나, 비주거적인 사용자 요구들을 처리하도록 한다. 시스템 내에는 [3]에서와 마찬가지로 사용자 요구-큐에 의해 두개의 추가적인 리스트가 필요



(a) 일반 디스크 배열



(b) 논리적으로 분할된 디스크 배열

→ 실제 데이터를 읽기위한 디스크 탐색
 - - - - - 디스크 헤드 초기화때 필요한 디스크 탐색

그림 2. 단일 주기내 디스크 탐색 영역

하다. 하나는 예약-리스트로, 서비스 제공이 확정된 스트림들이 실제 디스크 참조를 하기 전까지 디스크 대역폭을 예약해 두는 리스트이고, 다른 하나는 서비스-리스트로 각 서비스 그룹별로 유지되는 리스트이다. 시스템 초기화시 서비스 그룹에 대한 분할 영역 할당과 시스템 진행시 사용자 요구의 처리는 다음과 같이 한다

- 각 디스크를 P개의 영역으로 나누고, 전체적으로 할당된 영역들이 왼쪽-대칭(Left-Symmetric) 형태가 되도록 서비스 그룹들에게 할당한다
- 새로이 도착하는 사용자 요구들은 모두 사용자 요구-큐에 저장하며, 가용 대역폭이 생긴 서비스 그룹이 발생하면 다음과 같이 행한다
 - 해당 서비스 그룹의 영역내에 저장된 미디어를 요구하는 사용자 요구를 사용자 요구-큐에서 찾는다
 - 만일 상응하는 사용자 요구가 있다면, 해당 서비스 그룹의 영역의 경계선과 가장 인접한 사용자 요구를 찾는다.
 - 새로운 스트림을 예약-리스트에 넣고, 가용 대역폭을 예약한다.
- 예약-리스트내에 속한 스트림들을, 요구하는 비디오 화일의 첫 블록이 저장된 디스크와 자신이 예약한 서비스 그룹이 이번 주기에 이용하고자 하는 디스크와 일치하는 스트림은 서비스 제공이 시작된다.

그림 2는 8개의 디스크로 구성된 디스크 배열에서, 일반적인 경우와 4개의 영역으로 분할한 경우에 각 서비스 그룹들이 소요하게 되는 디스크 탐색 구간들을 보여주고 있다. 여기서 경선은 CSCAN 방식으로 스트림들을 처리하기 위해 디스크 헤드를 초기화하기 위한 최대 디스크 탐색 구간을 나타내고, 실선은 실제 데이터를 읽어들이면서 이동하게 되는 디스크 탐색 구간을 나타낸다. 그림 2(a)는 [3]에서 설명된 일반적인 서비스 그룹들의 디스크 탐색 구간을 나타내는데, 이들은 절근하고자 하는 위치에 상관없이 스트림들을 수용하므로 디스크 탐색 시간은 항상 최대 디스크 탐색 시간을 가중하여야 한다. 그림 2(b)는 4개의 영역으로 분할된 형태의 디스크 배열로서, 그림 2(a)에서와는 달리 각 서비스 그룹이 단일 주기 내에 데이터를 읽어들이기 위한 디스크 탐색 구간은 분할된 크기의 영역만큼으로 줄어들게 된다. 또한 이전 서비스 그룹에 의해 마지막으로 참조된 블록의 위치와 현재 서비스 그룹이 처음으로 읽어들이고자 하는 블록의 위치가 서로 인접하므로 초기화를 위한 디스크 탐색 시간도 짧아진다. 단일 디스크 입장에서 디스크 헤드 이동 거리를 살펴 보면, 일반적인 형태에서는 4주기 동안 8번의 최대 디스크 탐색 시간을 소요하게 되지만, 논리적으로 분할된 형태에서는 4주기 동안 2번의 최대 디스크 탐색 시간을 소요하게 된다. 분할 정도를 P라 하였을 때, 각 서비스 그룹이 수용할 수 있는 스트림의 수는 다음과 같이 표현될 수 있다. 먼저, 그림 2에서 서비스 그룹 0, 1, 2, 4, 5 와 6에 대해선

$$q \left(\frac{d}{r_{disk}} + t_{rot} + t_{settle} \right) + \frac{t_{seek}}{P} \leq \frac{d}{r_{disk}} \quad (2)$$

으로 계산되어 질 수 있고, 서비스 그룹 3과 7에 대해선

$$q \left(\frac{d}{r_{disk}} + t_{rot} + t_{settle} \right) + \frac{P+1}{P} t_{seek} \leq \frac{d}{r_{disk}} \quad (3)$$

과 같이 계산되어 질 수 있다. 단일 디스크 입장에서 보면 주기의 길이가 P배 만큼 길어진 것처럼 보여질 수 있지만, 실제 읽어들이는 블록의 크기는 변하지 않았으므로, 주기의 변화가 있는 것이 아니다

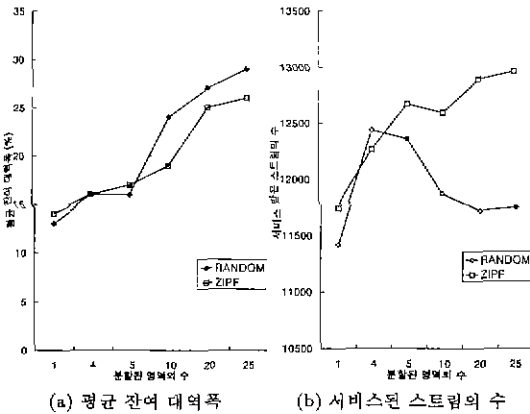


그림 3 실험 결과

5. 비디오 파일 배치

데이터들이 디스크들에 대해 고르게 분산된 디스크 배열에서, 사용자들의 데이터 참조가 모든 데이터들에 대해 고르게 분포한다면, 각각의 디스크 내부 트랙들도 고르게 참조될 것이다 하지만, 실제 주문형 비디오 서버로의 사용자 요구는 모든 비디오 파일들에 대해 고르게 분포되지 않고 Zipf 분포를 나타낸다 [5]. 따라서, 디스크 내부 트랙들에 대한 참조 분포도 Zipf 분포와 유사한 행태를 나타내게 된다.

이전 절에서 제시한 방식으로 디스크 배열을 분할하였을 경우, 사용자 요구가 Zipf 분포를 따르게 되면, 특정 서비스 그룹에 대해 사용자 요구가 집중되게 된다. 즉, 서비스 그룹들 간에 작업 부하가 불균형을 이루게 되어 시스템 전체적으로는 가장 대역폭이 있음에도 불구하고, 작업 부하가 심한 서비스 그룹에 속해야만 하는 사용자 요구들은 오랫동안 대기하여야 한다. 디스크 배열을 2개의 영역으로 분할한다고 할 때, 서비스 그룹들 간의 작업 부하의 균형을 유지하고, 분할에 의한 이득을 최대한 하기 위해 다음과 같이 비디오 파일들을 배치토록 한다.

- 비디오 파일들을 P개의 덩어리로 나누고, 각 덩어리 내 포함된 비디오 파일들의 참조 빈도수의 합이 비슷하도록 비디오 파일들을 분류한다. 이와 같이 분류된 덩어리들은 각 서비스 그룹들에 상응하게 되므로, 같은 영역 내에서는 참조 빈도의 차이가 있을 수 있지만, 서비스 그룹들 간의 작업 부하는 균형을 이루게 된다.
- 각 덩어리들이 다른 것들과 참조 빈도수는 비슷하지만 이들이 포함된 파일들의 수와 크기의 합은 다르므로, 디스크 배열은 논리적으로 분할할 때 균일한 크기로 분할하지 않고, 위에서 구해진 덩어리들의 크기 단위로 분할한다.

서비스 그룹들이 할당받은 영역내 데이터를 참조하는 스트림들만을 지원한다는 제약을 완화할 경우, 인접 영역에 대한 사용자 요구들을 포괄하는 경우가 자주 발생하여 분할에 의한 이득이 줄어들게 된다. 따라서, 각 서비스 그룹이 포함할 수 있는 영역의 범위는 사용자 요구의 참조 유형 변화에 따라 수정되어야 한다.

6. 실험

디스크 배열을 논리적으로 분할하여 얻어지는 이득을 모든 실험을 통하여 측정하여 보았다. 시스템은 용량이 2기가바이트인 디스크 [6] 100개로 구성하였으며, 디스크 시뮬레이션 부분은 비클리 대학에서 제작한 raidSim [7]이라는 RAID 시뮬레이터내 디스크 모듈을 사용하였다. 비디오 생성 시간은 75분에서 170분까지의 구간에서 평균 112분에 표준편차 20분인 정규분포를 따르는 것으로 하였다. 실험에서 주의 할 것은 초기 시작시 모든 사용자 요구들은 서비스 제공을 허가 받게 되고, 이후 가장 대역폭이 더 이상 없을 때부터 이들에 대한 서비스 제공이 끝날 때까지 긴 시간동안 서비스 제공은 거부될 것이고, 이러한 현상은 난수에 의한 사용자 요구 발생이 시스템을 일관된 상태로 만들 때까지 여러번 반복될 것이라는 것이다 [5]. 이러한 현상을 피하기 위해 본 실험에서는 초기에 모든 대역폭을 임의의 사용자 요구에 할당하고, 할당된 사용자 요구들은 잔여 생성시간이 0분에서 100분이 되도록 하였다. 또한 난수 발생으로 인한 효과가 발생할 때까지를 초기화 시간으로 정하고 이 시간 동안에는 통계 자료를 기록치 않도록 하였다. 실험은 초기화 시간 1만 초와 실험 시간 4만 초로 하여 총 5만 초에 대해 실시 하였다. 그 결과는 그림 3에 나타나 있다. 두 그래프에서 가로축은 분할된 영역의 수를 나타내는데, 분할된 영역의 수가 1인 경우는 일반적인 디스크 배열에 해당한다. 여기서 RANDOM과 ZIPF은 사용자들의 비디오 화일 참조 유형

을 나타내는데, RANDOM의 경우, 사용자이 참조 유형이 예측과는 달리 고른 분포로 비디오들을 참조하는 경우를 의미하며, ZIPF는 예측된 사용자 선호도와 실제 사용자들의 참조 유형이 모두 Zipf분포를 따르는 경우를 의미한다. 본 실험에서는 모든 서비스 그룹들이 자신이 할당받은 영역만을 서비스하도록 하여, 예측된 선호도와 실제 사용자 참조 유형에 따른 성능 차이 정도를 보일 수 있도록 하였다. 그림 3(a)는 모든 서비스 그룹들이 동일한 수의 스트림들을 지원한다고 가정하였을 때, 매 주기마다 남겨지는 대역폭의 정도를 백분율로 나타낸 것이다. 그림 3(b)는 비디오 화일들을 선호도에 기반하여 디스크 배열에 저장한 상태에서 디스크 유휴 정도에 따라 지원되는 스트림들의 수를 나타낸다. 이 둘을 비교하여 보면, 분할된 영역의 수가 1인 경우와 4인 경우에 있어서, 잔여 대역폭의 변화에 비해 서비스된 스트림의 수는 큰 차이를 보임을 알 수 있다. 이는 스트림 추가를 고려할 때, 분할된 영역의 수가 1인 경우엔 서비스 그룹이 처리하는 영역이 넓어 디스크 탐색 시간의 증가폭이 크지만, 분할된 영역의 수가 4이상일 경우에는, 디스크 탐색 시간의 증가폭이 작아 스트림이 추가될 수 있는 확률이 높기 때문이다. 그림 3을 살펴보면, 사용자 참조 유형이 비디오 화일 저장시 사용되었던 선호도를 따르는 경우엔 서비스된 스트림들의 수가 꾸준히 증가되는 반면, 사용자 참조 유형이 고른 분포를 나타내는 경우엔 분할된 영역의 수가 증가할 경우 일부 감소함을 볼 수 있다. 이는 각 서비스 그룹들이 자신이 할당받은 영역만을 서비스 하므로, 영역내 포함된 비디오 화일 수의 차이에 의해 발생한 서비스 그룹들 간의 작업 부하의 불균형에 의한 것이다. 이를 제거하기 위해선 시스템 동작시 서비스 그룹들이 할당 받은 영역들의 범위를 새로이 발생하는 사용자 요구 유형에 따라 동적으로 변화시켜 주어야 하는데, 이는 향후 연구 과제로 남긴다.

7. 결론

멀티미디어 서버를 설계함에 있어 가장 중요한 것은 주어진 자원을 가지고 얼마나 많은 사용자들에게 지속적인 서비스를 제공할 수 있는냐 하는 것이다. 디스크 배열을 이용하는 서버의 경우 블록의 크기를 증가시켜 주기의 길이를 늘임으로써 지원 가능한 스트림의 수를 증가시킬 수 있다. 하지만, 이와 같은 방법은 메모리 요구량 또한 현저하게 증가시키게 되어 시스템 전반에 걸친 설계 비용이 상승하게 된다. 본 논문에서는 이러한 서버에 대해 메모리 요구량 증가를 최소화 하면서 지원되는 스트림의 수를 늘릴수 있는 논리적 디스크 분할 기법을 제시하였다. 이는 디스크 탐색으로 낭비되는 시간을 줄여 같은 주기내에 보다 많은 스트림을 지원하는 것으로, 모든 실험 결과, 지원 가능한 스트림의 수가 평균 3~11% 증가하였고, 주기내 남겨지는 대역폭의 양도 평균 15% 이상이 되어 비동기적인 사용자 요구의 처리가 용이함을 보였다. 이는 [3]에서 제안된 시작 지연 시간을 줄이는 CAB 알고리즘과 혼용이 가능하며, 기존의 RAID 시스템과 같이 절한 허용을 위한 패리티 정보의 수용도 가능하다는 향후, 절할 허용 여부의 사용자 상호 작용에 관한 부분에 대한 추가적인 연구가 요구된다.

참고 문헌

- [1] D. Gemmell, H. Vin, D. Kundlur, P. Rangan, and L. Rowe, "Multimedia Storage Servers: A Tutorial and Survey," *IEEE Computer*, vol. 28, no. 5, pp. 40-49, May 1995.
- [2] P. Shenoy and H. M. Vin, "Efficient striping techniques for multimedia file servers," in *Proceedings of the 7th International Workshop on NOSSDAV*, pp. 25-36, May 1997.
- [3] B. Ozden, R. Rastogi, and A. Silberschatz, "Disk Striping in Video Server Environments," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, June 1996.
- [4] M. Seltzer, P. Chen, and J. Ousterhout, "Disk Scheduling Revisited," in *Proceedings of the 1990 Winter USENIX*, pp. 313-324, January 1990.
- [5] A. Dau, D. Sitaram, and P. Shahabuddin, "Scheduling Policies for an On-Demand Video Server with Batching," in *Proceedings of the 2nd Annual ACM Multimedia Conference and Exposition*, pp. 15-23, October 1994.
- [6] Quantum Corporation, "Quantum Viking Online Specifications," <http://www.quantum.com/products/hdd/viking/>, June 1998.
- [7] E. K. Lee, "Performance Modeling and Analysis of Disk Arrays," Tech. Rep. CSD-90-573, Dept. of Computer Science, University of California at Berkeley, 1990.