
사용자등급을 고려한 비디오 데이터베이스의 저장시스템

김근형* · 김두경**

Storage System of Video Database of Considering User Classes

Keunhyung Kim* · Doogyung Kim**

요약

비디오 데이터베이스는 비디오 정보를 저장하고 있다가 사용자가 요구한 비디오 정보를 검색해 주는 기능을 갖는다. 따라서, 비디오 데이터베이스의 목적은 비디오정보를 저장하고 검색하기 위한 편리하고도 효율적인 환경을 제공해야 할 뿐만 아니라 보다 많은 사용자들에 대해서 보다 신속하게 요구결과들을 검색해 줄 수 있어야 한다. 그러나, VOD응용과 같은 비디오 데이터베이스의 사용자들은 1시간이나 2시간 정도의 비디오 시청을 위하여 어느 정도의 시작지연은 감수할 수 있다.

본 논문에서는 비디오 데이터베이스의 사용자등급을 구분하여 효율적으로 서비스할 수 있는 비디오 데이터베이스의 하부 저장시스템을 제안한다. 제안하는 저장시스템은 다중 디스크 환경하에서 급박한 등급의 사용자들에게는 디스크헤드의 임의이동을 허용하고 VOD응용과 같은 덜 급박한 등급의 사용자들에게는 디스크헤드의 순차적 이동을 유지함으로써 급박한 사용자들에게 빠른 응답을 제공할 뿐만 아니라 전체적으로 보다 많은 사용자들을 지원할 수 있다.

ABSTRACT

Video database has the function that stores videos and retrieves the video informations for the service request of users. Therefore, the purpose of the video database is to provide not only an environment both convenient and efficient for retrieving and storing video, but also to serve the requested video informations with rapid startup latency for more users. But, some users of video database as VOD application might tolerate slow startup latency because they would watch video during one hour or two hours.

In this paper, we propose the novel policy of the storage system of video database, which serves efficiently the users by considering the classes of the users in video database. The policy admits the random movements of disk heads for urgent user classes but maintains the sequential or restricted movements of disk heads for less-urgent user classes like VOD application in storage system of video database. So, it can not provide only rapid startup latencies for urgent requests but also it can process more service requests in all.

*제주대학교 경영정보학과 전임강사
접수일자 : 2001. 9. 27

**제주대학교 경영정보학과 교수

1. 서론

비디오 데이터베이스시스템(Video DataBase System, VDBS)은 비디오 정보를 저장하고 있다가 사용자가 요구한 비디오 정보를 검색해 주는 기능을 갖는다. 따라서, VDBS의 주요 목적은 비디오 데이터들을 쉽게 저장할 수 있게 하는 것과 저장된 비디오 데이터들 중에서 원하는 부분을 신속하게 검색할 수 있게 하는 것, 보다 많은 사용자들을 서비스할 수 있게 하는 것이다[1]. 저장장치의 발전과 함께 비디오 데이터의 디지털화 및 압축기술의 발전은 VDBS의 편리한 저장기능을 가능하게 하였다. 원시 비디오데이터(raw video data)에 대한 메타 데이터(meta data) 개념은 저장된 비디오 데이터들 중에서 원하는 부분을 편리하고도 신속하게 검색할 수 있게 한다[2]. VDBS의 하부 저장시스템의 관점에서 볼 때 원하는 비디오 데이터 부분을 신속하게 검색하기 위한 또 하나의 요구사항은 디스크 헤드의 임의적 이동을 허용하는 것이다. 일반적인 디스크 시스템은 디스크헤드의 임의적 이동을 통하여 다중 서비스를 수행한다. 반면, VDBS가 동시에 보다 많은 다중 사용자들을 서비스하기 위해서는 하부 저장시스템의 제한된 디스크 대역폭을 효율적으로 활용하는 기술이 필요한데 이러한 기술은 디스크헤드의 순차적 이동을 통하여 다중 서비스 사이의 탐색지연시간을 절약하는 것이다[3]. 그러나, 디스크헤드를 순차적으로 이동시키면서 다중 사용자들을 서비스하려면 원하는 비디오 데이터의 신속한 검색이 불가능해진다. 따라서, VDBS의 성능을 보다 향상시키기 위하여, 하부 저장시스템의 디스크 대역폭을 효율적으로 활용하면서 서비스 요구들을 신속하게 검색해 줄 수 있는 방안이 필요하다.

VDBS의 사용자들 중 게임 S/W 등과 같은 것들은 서비스 요구에 대한 신속한 검색 결과가 필수적일 수 있지만[9] VOD응용은 1시간이나 2시간동안의 비디오 재생을 위하여 어느 정도의 시작지연은 감수할 수 있다. 본 논문에서는 VDBS 사용자들의 이러한 특성들을 바탕으로 VDBS 사용자들을 급박한 서비스요구와 덜 급박한 서비스요구로 분류하고 급박한 서비스요구들에게는 하부 저장시스템의 디스크헤드들에게 임의적 이동을 허용하여 서비스할 수 있게 하고 덜 급박한 서비스요구들에게는 순차적 이동을 통하여 서비스함

으로써 VDBS의 성능을 향상시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 비디오 데이터베이스에 대해서 설명하고 3장에서는 비디오 데이터베이스의 하부 저장시스템을 고찰하고 기존의 관련 연구들 및 문제점을 분석한다. 4장에서는 VDBS의 성능을 향상시킬 수 있는 새로운 하부 저장시스템을 제안한다. 5장에서는 모의실험을 통하여 제안된 정책을 평가하고 분석하며, 마지막으로 6장에서 결론을 맺는다.

II. 비디오 데이터베이스

VDBS의 주요 목적은 비디오 정보를 저장하고 검색하기 위한 편리하고도 효율적인 환경을 제공하는 것이다[1]. 좀 더 구체적으로 말하면, 순차적인 비디오 데이터의 특정 부분에 대한 임의접근을 제공하기 위하여 비디오 스트림을 의미있는 데이터단위들로 분할하여 각 데이터단위들을 인덱싱(indexing)하고 보다 쉬운 브라우징(browsing)과 검색(retrieval)을 할 수 있게 그 인덱스들을 표현하는 것이다[1]. 그리하여 VDBS는 원시 비디오데이터들에 대한 인덱스들의 데이터베이스라고 할 수 있다. 이러한 것들과 더불어 VDBS는 전통적인 DBMS(DataBase Management System)의 요구사항들, 즉 데이터 공유성, 보안성, 데이터 독립성, 중복배제성, 일치성, 무결성 등의 부분들도 고려하여야 한다.

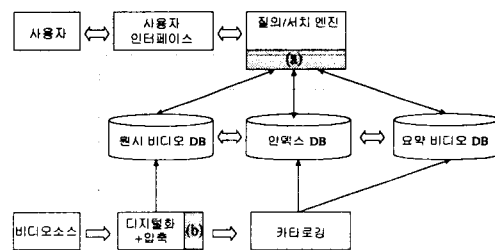


그림 1. VDBS의 블록다이어그램

그림1은 전형적인 VDBS의 블록다이어그램[4]을 나타내고 있다. 그림1에서 VDBS의 각 부분의 개략적인 기능명세는 다음과 같다.

- 사용자 인터페이스 : 편리하고도 시각적인 방법으로 데이터베이스와 상호작용하면서 (interactively) 질의(querying) 및 브라우즈(browse)할 수 있게 하고 선택된 비디오 클립(video clip)들을 볼수 있게 한다.
- 질의/서치 엔진(query/search engine) : 사용자가 제공한 파라미터들에 따라 데이터베이스를 서치한다.
- 원시 비디오DB : 디지털화되고 압축된 비디오데이터들의 저장소
- 요약 비디오DB : 간결하면서도 계층적인 방법으로 비디오 데이터들을 표현할 수 있는 요약된 비디오정보
- 인덱스DB : 비디오 데이터단위들에 대한 포인터들
- 디지털화+압축 : 아날로그 비디오데이터를 압축된 디지털 포맷으로 변환하기 위하여 필요한 하드웨어와 소프트웨어
- 카타로깅(cataloguing) : 원시 비디오데이터로부터 의미있는 데이터단위를 추출하고 대응하는 인덱스들을 생성 □

VDBS의 하부 저장시스템은 비디오데이터들을 저장장치에 저장하고 사용자요구에 따라 원시 비디오데이터의 어떤 부분을 읽어주는 기능을 갖는다고 할 수 있다. 이러한 관점에서 볼 때, 그림1에서 하부 저장시스템을 구성하는 부분은 「원시 비디오DB」와 「디지털화+압축」 그리고 「질의/서치 엔진에서 원시 비디오 DB에 접근하여 요구 비디오 데이터를 읽는 부분」이라고 할 수 있다. 「원시 비디오DB」를 위한 저장공간을 다중 디스크 환경이라고 가정할 때, VDBS에서 하부 저장시스템의 주요 기능은 비디오 데이터들을 다중 디스크상에 적절하게 배치하고 요구된 비디오 데이터들을 효율적으로 검색하는 것이라고 할 수 있다. 그림1에서 점선으로 둘러싸인 영역이라고 할 수 있다. VDBS에 도착한 서비스요구에 대하여 인덱스DB를 탐색하는 과정이 선행되고 여기에서의 결과를 바탕으로 하부 저장시스템은 원시 비디오DB를 검색하여 최종적인 결과를 도출한다. 다음 절에서 VDBS의 하부 저장시스템과 관련하여 기존에 연구된 주요 결과들을 고찰하고 분석해 본다.

III. 비디오 데이터의 저장 및 검색 정책

저장 시스템이 비디오 스트림에 물리적으로 접근하는 단위를 세그먼트라고 부른다. 즉, 스트림은 세그먼트들로 분할되어 다중 디스크들상에 분산 배치되고 스트림 검색시 세그먼트 단위로 읽어들인다[10]. 그림2는 비디오 스트림들 A, B, C가 각각 세그먼트들로 분할되어 디스크 D₀, D₁, D₂상에 배치된 상황을 나타내고 있다.

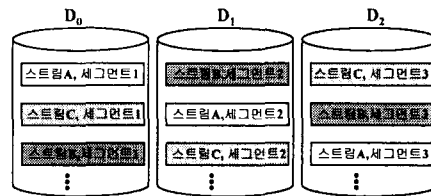


그림 2. 비디오스트림의 분산배치 모델

비디오 스트림을 검색하여 재생할 때는(서비스할 때) 그 스트림의 연속적인 세그먼트들을 세그먼트 단위로 하나씩 읽고 재생하는 과정을 반복한다. 서비스 주기는 하나의 세그먼트가 재생되는 시간이다. 다중 스트림들을 서비스하기 위하여 각 스트림 당 하나의 세그먼트씩을 서비스주기내에 모두 읽어들여야 한다. 그림3에서 나타내는 바와 같이 서비스주기(T_{period})동안 사용될 데이터를 자기에게 할당된 시간슬롯(time slot) 내에 모두 읽어들여야 한다. 그리하여, 현재 서비스를 받고 있는 스트림들은 종료되기까지 특정한 서비스주기를 두고 반복적으로 처리되어야 한다. 그림3은 단일 디스크 환경을 가정한 검색 서비스 모델이고 D_{latency}는 디스크 전 영역을 대상으로 어떤 스트림의 세그먼트를 읽기 위한 탐색지연시간이다.

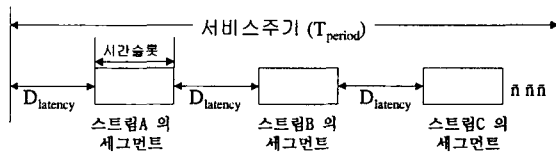


그림 3. 단일 디스크 환경에서의 다중 스트림 서비스 모델

서비스 주기는 세그먼트가 소모되는 시간보다 작아야 한다. 서비스 주기를 길게 하려면 세그먼트 크기를

크게 해야 한다. 서비스 주기가 세그먼트 소모시간 보다 길어지면 스트림 재생시 깜빡거림(hiccup) 현상이 일어난다. 따라서, 서비스 주기는 세그먼트 크기에 의하여 결정되고 식(1)을 만족해야 한다. 이후부터 주로 사용될 기호들중에서 T_{period} 는 서비스주기, S 는 세그먼트 크기, R_{med} 는 스트림소모율, R_{disk} 는 디스크 전송률, $D_{latency}$ 는 디스크내에서의 탐색지연시간, m 은 서비스 주기동안 검색될 세그먼트 개수를 의미한다.

$$m * \frac{S}{R_{disk}} + D_{latency} * \frac{1}{2} * T_{period} \leq \frac{S}{R_{med}} \quad (1)$$

그림4는 다중 디스크 환경에서 비디오 스트림들이 분산배치되었을 때의 스트림 검색서비스 모델이다. 어떤 서비스주기의 임의의 시간슬롯에서 다중 디스크 환경의 각 디스크는 상이한 스트림들의 세그먼트를 검색한다. 그림4에서 시간슬롯 T(1)동안 디스크 D₀에서 스트림A의 세그먼트1을 읽는 동안에 디스크 D₁과 D₂에서는 다른 스트림들의 세그먼트들을 읽는다.

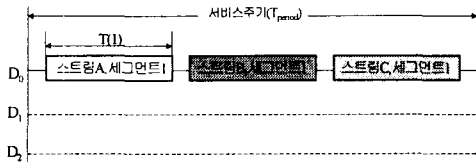


그림 4. 다중 디스크환경에서 다중 스트림 서비스 모델

다중 디스크환경에서 비디오 스트림의 저장 및 검색정책과 관련하여 기존에 연구된 주요 결과들은 RR(Round Robin), VSP(Virtual Sequential Pumping), GRP(Going and Returning Placement) 등이 있다. 각 정책들은 비디오 스트림 V_i 를 세그먼트들 $V_i^0, V_i^1, V_i^2, \dots$ 으로 분할하여 각 세그먼트들을 X개의 디스크들 $D_0, D_1, D_2, \dots, D_{X-1}$ 상에 분산 배치한다.

RR정책에서 임의의 비디오 세그먼트 V_{ij} 가 X개의 디스크들 $D_0, D_1, D_2, \dots, D_{X-1}$ 상에 배치될 때 디스크 $D_{(i-1) \bmod X}$ 의 임의의 위치에 배치된다[5]. RR정책은 각 디스크에서 디스크헤드의 임의이동에 기반하여 다중 세그먼트 서비스를 하므로 세그먼트 검색사이의 탐색 시간 등으로 디스크 대역폭을 낭비한다. 그러나, 스트림 요구에 대한 응답은 즉각적으로 이루어진다.

VSP 정책은 각 디스크 영역을 동일한 수의 블록데이터를 갖고 동일한 디스크 전송률을 갖는 서브영역

즉, 존(zone) 들로 분할하고 비디오 스트림의 세그먼트 들을 특정 디스크의 특정 존에 배치한다[3]. 디스크 배열에서 X개의 디스크 $D_0, D_1, D_2, \dots, D_{X-1}$ 이 있고 각 디스크는 $Y(=X+1)$ 개의 존들 $Z_0, Z_1, Z_2, \dots, Z_{Y-1}$ 로 나누어져 있다고 할 때 임의의 비디오 세그먼트 V_i^j 의 전역 세그먼트 번호가 g 라면 이 세그먼트는 디스크 $D_{g \bmod X}$ 의 존 $Z_{g \bmod Y}$ 에 배치된다. 전역세그먼트 번호는 세그먼트들에 부여된 일련번호인데 다른 스트림들의 세그먼트들도 고려하여 순서대로 부여된 세그먼트 번호이다. 예를 들어 스트림 V_0 와 V_1 의 세그먼트들에 대한 전역 세그먼트 번호를 구할 때 스트림 V_0 가 6개의 세그먼트들 $V_0^0, V_0^1, V_0^2, V_0^3, \dots, V_0^4, V_0^5$ 로 이루어질 경우 스트림 V_1 의 첫 번째 세그먼트 V_1^0 의 전역세그먼트 번호는 7이 된다. 데이터 검색 시 다중 디스크 환경의 모든 디스크들의 디스크헤드들은 서비스주기마다 존 단위로 동기화(synchronization)를 유지하면서 안쪽 존부터 바깥쪽 존으로 순차적 이동을 하면서 요구 세그먼트들을 읽는다. 특정 서비스주기동안에 각 디스크 들은 특정 존 내의 세그먼트들만을 검색하므로 탐색지연시간은 RR정책의 경우보다 짧아진다. RR정책의 탐색지연을 $D_{latency}$ 라고 했을 때 VSP의 탐색지연 $Z_{latency}$ 는 식(2)와 같이 표현될 수 있다.

$$Z_{latency} \leq \frac{D_{latency}}{Y} \quad (2)$$

(Y : 단일 디스크에서의 존 갯수)

$Z_{latency}$ 는 $D_{latency}$ 보다 짧으므로 VSP정책의 디스크 대역폭은 RR정책에 비해서 절약되고 절약된 디스크 대역폭만큼 더 많은 스트림들을 서비스할 수 있다. 그러나, VSP정책에서 시작 지연은 RR정책보다 더 길다. RR정책에서는 디스크 헤드의 임의이동을 기반으로 함으로 요구 스트림의 시작 세그먼트를 즉각적으로 읽을 수 있지만 VSP정책에서는 디스크 헤드가 다른 디스크들의 헤드들과 존 단위의 동기화를 유지하면서 요구 스트림의 시작 세그먼트가 있는 존으로 올 때 까지 기다려야 한다. 또한, 요구 스트림들의 세그먼트들이 특정 존 영역으로 집중될 경우 시작 지연은 더 길어질 수 있다.

GRP 정책은 VSP정책의 되감기 시간을 제거하기 위하여 제안되었다[6]. VSP정책이 세그먼트들을 배치할 때 하향방향(안쪽 존부터 바깥쪽 존 방향)으로만 배치하는데 반하여 GRP정책은 하향방향과 상향방향

(바깥쪽 존부터 안쪽 존 방향)으로 번갈아 가면서 배치한다. 그림5는 GRP 배치정책을 수식으로 나타내고 있다. 그림5에서 g 는 배치될 세그먼트의 전역세그먼트 번호를 의미하고 X 는 디스크 수, Y 는 각 디스크의 존 개수이다. 전역세그먼트 번호가 g 인 세그먼트는 디스크 D_0 의 존 Z_2 에 배치된다.

$$C_{topdown} = \left(\left\lfloor \frac{g}{Y} \right\rfloor + 1 \right) \bmod 2$$

$$C_{bottomup} = \left(\left\lfloor \frac{g}{Y} \right\rfloor \right) \bmod 2$$

$$d = (g \bmod X)$$

$$z = C_{topdown} * (g \bmod Y) + C_{bottomup} * (Y - 1 - g \bmod Y)$$

그림 5. GRP 배치 정책

그림6은 전역세그먼트 번호가 0부터 19까지인 20개의 세그먼트들이 VSP 배치정책에 의하여 배치된 상태와 GRP에 의하여 배치된 상태를 나타내고 있다. 그림6에서 볼 수 있는 것처럼 VSP는 하향방향으로만 세그먼트들을 배치하고 있는데 반하여 GRP는 하향방향과 상향방향을 번갈아 가면서 다중 디스크 상에 배치하고 있다. 디스크 헤드의 순차적 이동을 유지하기 위하여 하향방향으로 배치된 세그먼트는 세그먼트 검색시에도 디스크헤드가 하향방향으로 움직일 때 검색되어야 하고 상향방향으로 배치된 세그먼트는 세그먼트 검색시 디스크헤드가 상향방향으로 움직일 때 검색되어야 한다.

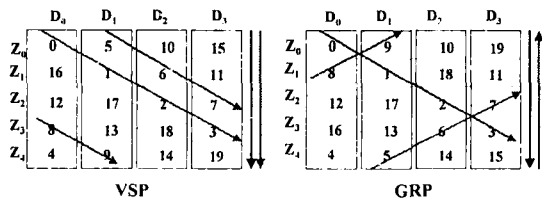


그림 6. VSP배치과 GRP배치정책의 비교

디스크 헤드의 순차적 이동을 유도하기 위하여 GRP 배치정책을 따르는 단순한 검색정책은 존 사이의 동기화를 유지하면서 디스크 헤드를 존 Z_0 부터 존 Z_4 까지 하향방향으로 움직이면서 세그먼트 검색서비스를 하고 다시 존 Z_4 부터 Z_0 으로 이동하면서 상향방향으로 세그먼트 검색 서비스를 하는 과정을 반복한다. 그러나, 이러한 단순한 방식은 임의의 서비스주기에서 다중 디스크 환경의 디스크들중 1/2 이상을 활용

할 수 없게 할 뿐만 아니라 시작 지연 시간도 VSP정책의 최대 2배까지 길어질 수 있다.

N. 새로운 검색정책

VDBS의 사용자들 중 게임 S/W 등과 같은 것들은 서비스 요구에 대한 신속한 검색 결과가 필수적일 수 있지만 VOD응용은 1시간이나 2시간 정도의 비디오 재생을 위하여 어느 정도의 시작지연은 감수할 수 있다. 앞에서 살펴 본 RR정책은 게임 S/W와 같은 급박한 서비스 요구에 적합하고 VSP정책은 비디오 시청자와 같은 덜 급박한 서비스 요구에 합당한 정책이다. 여기서 제안하는 정책은 급박한 서비스요구와 덜 급박한 서비스요구들이 혼합해서 도착하는 상황에 적합한 정책이다. 즉, 다중 디스크 환경의 디스크들 중 1/2은 RR정책을 따르고 나머지 1/2은 VSP정책보다 성능이 우수한(세그먼트사이의 탐색지연시간만을 고려할 때) GRP(Going and Returning Placement) 정책을 따른다. 기존의 VSP정책이 서비스주기동안 퇴감기 시간으로 인하여 디스크 대역폭을 낭비했지만 GRP정책은 퇴감기 시간을 제거함으로써 VSP정책보다 세그먼트 사이의 평균 탐색지연시간이 더 짧다.

그림7은 제안하는 검색정책의 개념을 도식적으로 나타내고 있다. 그림7에서 볼 수 있듯이 임의의 서비스 주기동안 어떤 디스크들은 디스크헤드의 동기화 및 순차적 이동(즉, 존 내에 있는 세그먼트들만 검색)을 유지하면서 세그먼트 검색 서비스를 하고 어떤 디스크들은 디스크헤드의 임의적 이동에 의하여 세그먼트 검색 서비스를 한다. 급박한 서비스요구들에게는 임의접근 서비스를 제공하고 덜 급박한 서비스요구들에게는 순차접근 서비스를 제공함으로써 사용자들의 다양한 요구를 만족시킬 수 있다. 제안하는 검색정책은 전체 서비스 가능한 사용자들 중의 약 50% 정도에게는 RR 정책만큼 빠른 응답시간을 제공할 수 있고 수용 가능한 스트림들의 수도 RR정책보다 많게 한다.

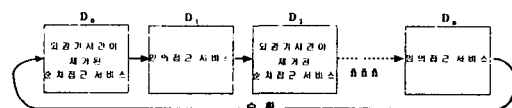


그림 7. 새로운 검색정책 개념

예제를 통하여 제안하는 검색정책을 세부적으로 고찰하기로 한다. 그림8은 전역세그먼트 번호가 0부터 19까지인 20개의 세그먼트들이 GRP 배치정책에 의하여 배치된 상태와 각 세그먼트들의 배치방향을 나타내고 있다. 또한, 임의의 서비스주기동안에 각 디스크에서 세그먼트 검색 서비스를 하는 형태를 나타내고 있다. 그림8에서 화살표 ↓은 하향방향으로 배치된 세그먼트를 의미하고 화살표 ↑은 상향방향으로 배치된 세그먼트를 의미한다. T_i 는 i 번째 서비스 주기를 의미한다.

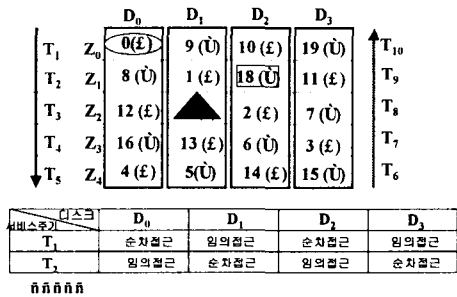


그림 8. 제안된 검색정책의 예

GRP 배치정책에 의하여 디스크 D_0 의 존 Z_0 에는 하향방향의 세그먼트들만 존재하고 디스크 D_0 의 존 Z_1 에는 상향방향의 세그먼트들만 존재한다. 즉, 화살표 ↓로 표시된 존에는 하향방향의 세그먼트들만 존재하고 화살표 ↑로 표시된 존에는 상향방향의 세그먼트들만 존재한다. 단순한 GRP 검색정책의 경우, 디스크 헤드는 존 Z_0 부터 존 Z_4 까지 하향방향으로 움직이면서 세그먼트 검색서비스를 한다고 할 때(이때, 각 디스크헤드는 존 사이의 동기화를 유지한다) 서비스주기 T_1 동안 디스크 헤드들이 디스크 D_0 의 존 Z_0 영역과 디스크 D_2 의 존 Z_0 영역에서는 세그먼트 검색서비스를 정상적으로 할 수 있지만 디스크 D_1 과 D_3 에서는 유휴 상태로 남게 된다. 왜냐하면, 디스크 D_1 의 존 Z_0 와 디스크 D_3 의 존 Z_0 에는 상향방향의 세그먼트들만 존재하기 때문이다.

제안하는 검색정책에서는 서비스주기 T_1 동안에 디스크 D_1 과 D_3 에 임의접근을 허용하여 유휴상태로 남아있지 않게 한다. 예를 들어, 서비스 주기 T_1 동안에 시작 세그먼트의 전역 세그먼트 번호가 0인 스트림1 서비스 요구(그림8에서 타원형으로 둘러싸인 세그먼트,

VOD응용과 같은 덜 급박한 요구라고 가정)와 시작 세그먼트의 전역 세그먼트 번호가 17인 스트림2 서비스 요구(그림8에서 삼각형으로 둘러싸인 세그먼트, 게임 S/W와 같은 급박한 요구라고 가정)가 도착했을 때 디스크 헤드의 순차적 이동을 유지하기 위해서는 디스크 D_1 이 유휴하게 되므로 T_1 동안에 디스크 D_1 에서는 디스크헤드의 임의이동에 의한 세그먼트 검색 서비스를 허용함으로써 디스크 D_1 을 유휴하게 남겨두지 않을 뿐만 아니라 빠른 응답시간을 제공할 수 있다. 다음 서비스 주기 T_2 에서 디스크 D_1 은 존 Z_1 영역에서 순차접근에 의해서 세그먼트 1을 검색함으로써 스트림1 서비스를 지속하게 되고 디스크 D_2 는 디스크 헤드의 임의적 이동을 허용함으로써 세그먼트 18을 검색하고 스트림2 서비스를 지속하게 된다. 즉, 각 디스크들은 매 서비스 주기마다 순차접근 모드(mode)와 임의접근 모드를 교체하면서 서비스를 한다.

제안하는 검색정책에서 또하나 고려할 사항은 시작 지연시간이다. 그림8에서, 서비스 주기 T_1 동안에 시작 세그먼트의 전역 세그먼트 번호가 7인 스트림 3 서비스 요구(VOD응용과 같은 급박하지 않은 요구라고 가정)가 도착했을 때 디스크 헤드의 순차적 이동을 유지하기 위해서 현재 서비스 주기가 T_8 이 될 때까지 「8*서비스주기」 동안 기다려야 한다. 이는 VSP정책에서의 최대 시작지연시간(그림8의 경우 「5*서비스주기」)보다 더 길다. 따라서, 제안하는 검색정책에서는 시작지연을 단축시키기 위하여 디스크의 존 개수를 VSP의 1/2배로 감소시킨다. 존 개수가 감소되면 최대 시작지연시간은 짧아지지만 존 영역이 넓어지고 존 내에서의 탐색지연 시간도 그만큼 길어지므로 허용할 수 있는 스트림 수도 VSP정책 보다 적어지리라고 생각할 수 있다. 그러나, 제안하는 검색정책은 각 서비스 주기에서 되감기시간을 제거함으로써 존 영역이 VSP보다 2배 넓어진다고 할 지라도 존 영역 내에서의 평균 탐색지연시간은 VSP정책보다 짧다. 그림9는 이에 대한 개념을 도식적으로 나타내고 있다.

그림9에서 VSP정책의 경우, T_1 동안에 시작 세그먼트가 7인 급박하지 않은 스트림 요구가 왔을 때 순차적 이동을 유지하기 위하여 현재 서비스주기가 T_8 이 될 때까지 기다려야 한다. 제안하는 정책에서 디스크의 존 개수가 VSP와 같은 8개이면 T_1 동안에 도착한 시작 세그먼트가 7인 스트림 요구를 서비스하기 위하

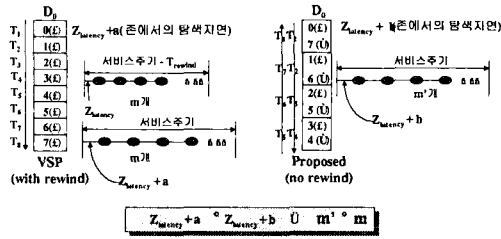


그림 9. 되감기시간 제거의 효과

여 현재 서비스주기가 T_{16} 이 될 때까지 기다려야 하지만 존 개수를 VSP의 1/2배로 축소시킴으로써 덜 급박한 스트림 요구에 대한 시작지연은 VSP정책과 동일하게 유지할 수 있다. 이때, 제안하는 정책의 존 영역은 VSP정책보다 2배 넓어서 존 내에서 세그먼트 검색 사이의 탐색지연은 길어지지만 되감기 시간이 서비스주기 내에 포함되지 않으므로 실질적인 평균 탐색지연 시간은 VSP정책보다 짧아지게 된다.

V. 실험 및 결과 분석

이 장에서는 제안한 검색정책을 동시 허용 스트림 수와 평균 시작지연 시간을 기준으로 RR 및 VSP정책과 비교하여 성능평가 및 분석을 한다. 실험을 위한 몇 가지 가정을 설정하였다. 첫째, 다중 디스크 환경의 각 디스크 헤드들은 하드웨어적으로는 비동기적으로 움직일 수 있다. 현재 상용화된 RAID시스템의 각 디스크들은 독립적인 디스크 컨트롤러들이 장착되어 있는 것이 일반적이므로 이러한 가정은 무리가 없을 것으로 판단된다[7]. 둘째, 임의의 수의 비디오 스트림들은 MPEG-I으로 압축되어 관련 배치 정책을 기반으로 다중 디스크상에 배치되어 있다고 가정하고 랜덤함수를 사용하여 요구 스트림의 시작 세그먼트가 있는 디스크와 존이 결정된다. 셋째, 요구 스트림의 도착시간은 평균이 3초인 Poisson분포를 따른다. 넷째, 시작지연 시간은 요구 비디오데이터의 위치를 인덱스DB로부터 계산한 후부터 요구비디오의 첫번째 세그먼트를 읽을 때까지의 시간이다. 즉, 인덱스DB를 탐색하는 시간은 시작지연시간으로부터 제외된다. 다섯째, 시작지연 시간 측정 시 동시에(concurrently) 허용 가능한 스트림 수까지의 평균 시작지연시간만을 고려한다. 저장 시스템이 최대 허용 가능한 스트림 수를 서비스하는 상

황에서 추가적으로 도착한 스트림 요구들의 시작지연 시간은 시스템 성능 요인보다는 사용자의 상태 및 반응에 의해서도 영향을 받을 것이기 때문이다. 실험을 위한 디스크 모델은 Seagate Wren8 ST41650N[8]를 사용한다.

시뮬레이터(Simulator)는 전역 스케줄러와 지역 스케줄러들, 큐들로 이루어진다. 지역 스케줄러와 큐는 각 디스크마다 하나씩 할당된다. 전역 스케줄러는 스트림 서비스요구의 허용여부를 결정하고 급박한 서비스요구와 덜 급박한 서비스요구를 분류하여 적절한 세그먼트요구들을 지역 스케줄러의 큐들에 할당한다. 지역 스케줄러는 자신의 큐에 있는 세그먼트요구들을 서비스한다. 시뮬레이터는 Visual SLAM의 PC 버전인 AweSim2.0과 C언어를 이용하여 구현되었고 펜티엄III의 윈도우98 환경에서 수행된다.

이상과 같은 실험 환경하에서 시뮬레이션은 다중 디스크 환경을 변화시켜 가면서 동시 허용 스트림 수 및 평균 시작지연시간을 측정한다.

그림10은 각 정책에 대하여 다중 디스크 환경의 디스크 수를 변화시켰을 때 디스크 수당 최대의 동시 허용스트림 수를 보여주고 있다. 동시에 허용 가능한 스트림 수의 관점에서 제안한 정책은 RR보다 성능이 우수하고 VSP와 비슷함을 알 수 있다.

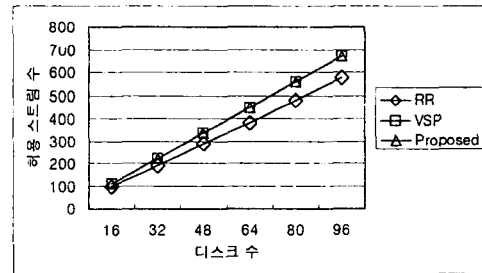


그림 10. 디스크 수 당 허용 스트림 수

그림11은 각 정책에 대하여 다중 디스크 환경의 디스크 수를 변화시켰을 때 디스크 수당 평균 시작지연 시간을 보여주고 있다. 그림11에서 Proposed(1)은 급박한 서비스 요구에 대한 검색정책을 의미하고 Proposed(2)는 덜 급박한 서비스 요구에 대한 제안한 검색정책을 의미한다. 시작지연의 관점에서 Proposed(1)은 RR보다는 성능이 약간 떨어지는데 이

는 각 디스크에서 필요한 경우 순차접근 모드로부터 임의접근 모드로 변환하는데 소요되는 오버헤드에 기인한다. 마찬가지로 Proposed(2)는 VSP보다는 성능이 떨어지는데 이는 임의접근 모드로부터 순차접근 모드로 변환하는데 소요되는 오버헤드에 때문이다. 그러나, 제안한 정책은 VSP만큼의 사용자 수들을 지원하면서 급박한 서비스 요구에 대해서는 RR과 거의 비슷한 시작지연을 제공할 수 있고 덜 급박한 서비스 요구에 대해서는 VSP와 거의 비슷한 시작지연을 제공할 수 있다.

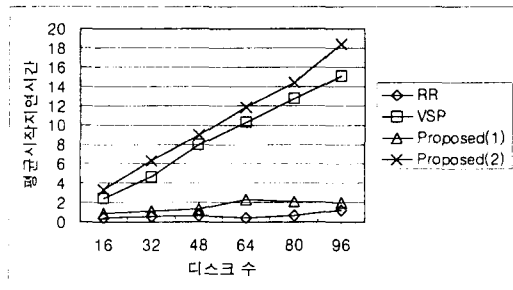


그림 11. 디스크 수당 평균 시작지연 시간

VI. 결론

비디오 데이터는 시각적 효과를 극대화할 수 있기 때문에 컴퓨터와 관련된 다양한 분야에서 상업적 잠재력이 매우 크다. 또한, 전세계적으로 많은 양의 비디오 데이터가 존재하고 생성되고 있기 때문에 이러한 비디오 데이터를 체계적이고 효율적으로 관리할 수 있는 VDBS의 필요성 및 중요성은 날로 증대되고 있다.

본 논문에서는 VDBS의 하부 저장시스템과 관련한 새로운 정책을 제안하였다. 제안한 정책은 사용자들의 신속한 검색 요구를 만족시키면서 VDBS가 지원할 수 있는 사용자들의 수를 많게 할 수 있는 방안이다. 즉, VDBS의 사용자들을 급박한 서비스요구들과 덜 급박한 서비스요구들의 두 부류로 구분하고 급박한 서비스요구들에 대해서는 디스크헤드의 임의이동을 허용하고 덜 급박한 서비스요구들에는 디스크헤드의 순차이동을 유도함으로써 급박한 서비스요구들에 대한 빠른 검색결과를 제공할 수 있을 뿐만 아니라 디스크 처리율을 높일 수 있다.

본 논문에서는 비디오 연산들 중 정속재생(norm

al-play)과 읽기전용트랜잭션(read-only-trans action)만을 고려하고 있지만 향후 고속재생(fast-play) 및 읽기쓰기 트랜잭션(read-write-transaction)을 위한 효율적인 하부 저장시스템에 관한 연구가 필요하다.

참고문헌

- [1] Oge Marques and Borko Furht, Issues in Designing Contemporary Video Database Systems, Proceedings of IASTED International Conference IMSA'99, Oct, pp43-47, 1999.
- [2] H.-J.Zhang, "Content-based video browsing and retrieval", in Handbook of Internet and Multimedia Systems and Applications, B.Furht(ed). Boca Raton:CRC Press,1999.
- [3] Gin-Kou Ma, Chiung-Shien Wu, Mei-Chian Liu, and Bao-Shuh P. Lin, Efficient Real-time DataRetrieval Through Saclable Multimedia Storage, Proceedings of ACM Multimedia 97, pp165-172, 1997.
- [4] B.-L. Yeo and M.M.Yeing,"Retrieving and Visualizing Video", Communications of the ACM, Vol.40, No.12, December 1997.
- [5] Taek-Geun Kwon, Sukho Lee, Data Placement for Continuous Media in Multimedia DBMS, Proceedings of IEEE Multimedia DBMS, August 28-30, pp110-117, 1995.
- [6] S.Park, KeunHyung.Kim, Efficient Storage and Retrieval Scheme of Video Streams in VOD Server, Proceedings of IMSA'99(in the IASTED International Conference), pp163-169, 1999,.
- [7] Patterson, D., Gibson, G., and Katz, R., A case for Redundant Arrays of Inexpensive Disks(RAID), Proceedings of SIGMOS88.
- [8] Huang-Jen Chen, Thomas D.C. Little, Storage Allocation Policies for Time-Dependent Multimedia Data, IEEE Transactions on Knowledge and Data Engineering, VOL.8, NO.5, OCTOBER, pp855-864, 1996.
- [9] Edward Chang and Hector Garcia-Molina, BubbleUp:Low Latency Fast Scan for Media

Servers, Proceedings of ACM Multimedia 97, pp87-98, 1997.

- [10] D.James Gemmel, Harrick M.Vin, Dilip D.Kandlur, P.VenkatRangan, Multimedia Storage Server: A Tutorial and Survey, IEEE Computer, Vol. 28, number 5, May, pp40-49, 1995.

김근형(Keunhyung Kim)

1990년 2월 서강대학교 컴퓨터학과(공학사)
1992년 2월 서강대학교 컴퓨터학과(공학석사)
2001년 2월 서강대학교 컴퓨터학과(공학박사)
1992년~1994년 현대전자 소프트웨어연구소
1994년~2001년 8월 제주산업정보대학 조교수
2001년~현재 제주대학교 경영정보학과 전임강사
관심분야 : 데이터베이스, 멀티미디어, 데이터마이닝, 데이터웨어하우스

김두경(Dugyung Kim)

1976년 2월 서강대학교 전자공학과(공학사)
1983년 8월 중앙대학교 전자공학과(공학석사)
1993년 2월 중앙대학교 전자공학과(공학박사)
1998년~현재 제주대 경영정보학과 교수
관심분야: 시스템분석, 정보통신