

# VBR 비디오 서버를 위한 확률 모델 기반의 최적 자원 구성

(An Optimal Resource Configuration Method based on  
Probability Model for VBR Video Server)

조대현<sup>\*</sup> 손진현<sup>\*</sup> 김명호<sup>\*\*</sup> 이윤준<sup>\*\*</sup>

(Dae Hyun Cho) (Jin Hyun Son) (Myoung Ho Kim) (Yoon Joon Lee)

**요약** 최근 대부분의 비디오 서버에서는 VBR 비디오를 사용하고 있고, VBR 비디오의 디스플레이 비율은 기존의 CBR 비디오의 디스플레이 비율과는 달리 시간에 따라 변하기 때문에, 기존의 CBR 기반의 자원 구성 방법을 VBR 비디오 서버의 자원 구성에 사용하기에는 적합하지 못하다. 본 논문에서는 확률 모델 기반의 효율적인 VBR 비디오 서버를 위한 자원 구성 방법을 제안한다. 이 방법은 비디오 서버의 처리량은 그대로 유지하면서 하드웨어 비용을 최소화시킬 수 있는 디스크와 메모리 양, 그리고 비디오 서버의 디스크 접근 주기 등과 같은 변수값들을 결정한다. 추가적으로 실험을 통하여 제안된 자원 구성 방법의 유용함을 확인한다.

**Abstract** Most of currently used videos have variable bit rate(VBR) characteristics. Since the display rate of VBR videos compared to CBR videos vary with time, it is not proper to configure resources of the VBR video server using the method proposed for the CBR video server. In this paper we propose an optimal resource configuration method for the VBR video server which is based on the probability model.

The proposed method decides the amount of disk and memory, and the disk access cycle of the video server with the lowest hardware cost, while preserving the throughput of the video server. In addition, we show the usefulness of the method through the various experiments.

## 1. 서론

최근의 고성능 컴퓨터 및 네트워크 환경의 발달로 사용자의 요구에 따라 뉴스나 영화 등과 같은 대용량 비디오 객체를 실시간으로 전송하는 주문형 비디오(VOD: Video-On-Demand) 서비스에 대한 관심이 증가되고 있다. 이러한 VOD 서비스를 지원하기 위해서는 비디오 객체들을 관리하고 전송하는 비디오 서버, 높은 네트워크 대역폭, 그리고 전송된 비디오 객체를 상영할 수 있는 비디오 클라이언트 등이 필요하다. 특히, 비디오 서

버는 서로 독립적으로 도착하는 다수의 사용자 요청들에 대해 비디오 스트림을 연속적으로 제공할 수 있어야 하므로 대용량 메모리와 디스크, 그리고 높은 디스크 전송률 등이 요구된다.

지금까지의 VOD 서비스에 대한 연구는 대부분 비디오 서버를 구성하는 자원들을 효율적으로 사용하는 방법들에 관하였다[1][2][3][4][5][6][7]. 대표적으로, 디스크 처리량을 향상시키는 기법으로는 디스크 스케줄링 방법(disk scheduling)[1][2][3]과 데이터 배치 방법(data placement)[4][5] 등이 연구되었으며, 비디오 서버의 효율적인 메모리 사용을 위해 고정 순서 I/O 스케줄링 방법(fixed-order I/O scheduling)과 버퍼 메모리 공유 기법(buffer sharing)[6][7]이 개발되었다. 그러나, 이들 방법들은 비디오 서버를 구성하는 여러 자원들의 상호 연관성은 배제하고 특정 자원의 관점에서 연구되었기 때문에, 비디오 서버 전체의 성능 향상에는 많은 제한이 있다[8]. 즉, 디스크의 처리량을 높이기 위해서

· 본 연구는 HV center의 지원으로 수행되었음.

† 비 회 원 : 한국과학기술원 전산학과  
dhcho@dbserver.kaist.ac.kr  
jhson@dbserver.kaist.ac.kr

\*\* 종신회원 : 한국과학기술원 전산학과 교수  
mhkim@dbserver.kaist.ac.kr  
yjlee@dbserver.kaist.ac.kr

논문접수 : 2000년 12월 8일

심사완료 : 2001년 5월 17일

는 보다 많은 메모리를 요구하게 되고, 메모리의 최적 사용 방법들은 디스크에서의 처리를 지연시키게 된다. 그러므로 특정 자원의 효율적인 사용 방법들을 고려하기 이전에 자원들 사이의 연관성을 규명하여 비디오 서버의 처리량을 극대화시킬 수 있는 최적의 자원 구성 방법의 개발이 필요하다. 특히, 디스크와 메모리는 비디오 서버의 성능 향상에 영향을 미치는 가장 중요한 자원들이므로 이들에 대한 면밀한 분석이 요구된다. 한번의 I/O에서 가능한 한 많은 양의 데이터를 읽음으로써 디스크 처리량을 향상시킬 수 있지만, 그 데이터를 모두 메모리에 버퍼링(buffering)하고 있어야 하므로 메모리 요구량은 상대적으로 늘어난다. 반면, 메모리 요구량을 줄이기 위해 한번의 I/O에서 읽어 들이는 데이터의 양을 작게 하면, 디스크 읽기 시간에 비해 디스크 지연 시간의 비중이 커져서 디스크 처리량은 급속히 떨어진다. 이러한 이유로 디스크와 메모리의 균형 있는 사용이 중요하며, 하드웨어 비용은 이를 결정할 수 있는 가장 중요한 기준이 될 수 있다. 일반적으로 비디오 데이터는 주기적으로 디스크에서 메모리로 전달되는데, 디스크에서 한번에 읽어 들이는 비디오 데이터의 양은 다음 주기까지 상영될 수 있는 충분한 양이어야 한다. 그러므로 비디오 데이터에 대한 디스크 접근 주기는 디스크에서 한번에 읽어 들이는 비디오 데이터의 양과 밀접한 관련이 있으며, 결국에는 디스크와 메모리의 크기에 영향을 미친다. 본 논문에서는 비디오 서버의 처리량은 유지하면서 디스크와 메모리의 하드웨어 비용을 최소화시킬 수 있는 비디오 서버의 디스크 접근 주기를 계산하고, 이를 기반으로 디스크와 메모리의 양을 결정하고자 한다.

상수 전송률인 CBR(Constant Bit Rate) 비디오 스트림을 다루는 비디오 서버에서 효율적인 디스크와 메모리 자원을 구성하는 방법은 이미 [9]에서 연구되었다. [9]에서는 디스크 이용도(disk utilization)와 디스크 지연 시간의 관계, 디스크 이용도와 요구되는 메모리의 관계에 대하여 설명하였다. 또한, 디스크와 메모리의 비용을 포함한 CBR 비디오의 스트림 당 비용 분석과, 이를 통한 전체 비용을 최소화하는 방안에 대하여 기술하고 있다. 그러나 현재 대부분의 비디오 서버에서 사용하는 비디오 객체는 가변 전송률인 VBR(Variable Bit Rate) 속성을 가지고 있다. CBR 비디오 스트림과는 달리 VBR 비디오 스트림은 시간에 따라 요구되는 비디오 서버 자원의 양이 일정하지 않기 때문에, VBR 비디오 서버의 자원을 구성하는데 있어서 [9]에서 제안된 CBR 비디오 서버 자원 구성 방법을 적용할 수 없다. 만약 간단히 VBR 비디오의 평균 디스플레이 비율을 사용하여

CBR 비디오 서버 자원 구성 방법으로 자원을 구성한다면, 자원 부족으로 좋은 품질의 서비스를 할 수 없는 경우가 발생할 수 있다. 또한 VBR 비디오의 최대 디스플레이 비율을 사용하여 CBR 비디오 서버 자원 구성 방법으로 자원을 구성한다면, 디스플레이 비율이 최대값이 아닌 대부분의 경우에서 많은 자원을 낭비할 수 있다. 이에 본 논문에서는 VBR 비디오의 특성을 분석하여 VBR 비디오 서버의 최적 자원 구성 방법을 제안한다. 그리고 실험을 통하여 제안한 방법의 효율성을 확인한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 고려하는 VBR 비디오 서버 모델에 대하여 언급한다. 3장에서는 제안하고자 하는 확률 모델 기반의 최적 자원 구성 방법에 대하여 설명한다. 그리고, 4장에서는 실험을 통하여 제안한 방법의 효율성을 확인한다. 마지막으로 5장에서는 본 논문의 공헌을 요약하고 앞으로 연구해야 할 분야를 정리함으로써 결론을 맺는다.

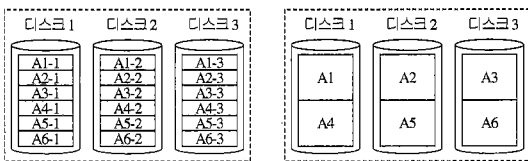
## 2. 비디오 서버 모델

비디오 서비스 시스템은 비디오 서버 부분과 클라이언트 부분으로 구분된다. 클라이언트들은 서로 독립적으로 비디오 서비스를 요구하며, 비디오 서버는 디스크에서 클라이언트가 요구한 비디오 데이터를 읽어 버퍼 메모리와 네트워크를 통하여 실시간으로 전송한다[10]. 비디오 서버 시스템이 관리하는 비디오 데이터는 영상 및 음성 정보를 포함하는 멀티미디어 데이터로서 많은 저장 공간을 필요로 하고 높은 디스플레이 비율을 요구한다. 이러한 비디오 데이터를 동시에 여러 클라이언트들에게 실시간으로 서비스하기 위해서, 비디오 서버는 대용량의 저장 공간과 높은 대역폭을 갖는 디스크를 필요로 한다. 일반적으로, 하나의 디스크로는 이러한 요구를 수용할 수가 없으므로, 독립된 컨트롤러를 가진 디스크들의 집합인 디스크 배열(disk array)을 사용한다[1][11][12][13][14]. 디스크 배열에서 하나의 디스크는 여러 개의 비디오 스트림을 동시에 지원할 수 있기 때문에, 각 디스크의 전송률은 클라이언트에서 비디오를 상영하기 위해 요구되는 최소 데이터 전송률인 디스플레이 비율보다 크다. 따라서, 디스크 컨트롤러는 한번에 여러 개의 비디오 데이터를 디스크 전송률로 읽어서 메모리에 저장하고, 네트워크 컨트롤러는 메모리에 저장된 비디오 데이터를 디스플레이 비율로 점차적으로 클라이언트에게 전송한다. 이러한 디스크와 네트워크의 전송 방식의 차이 때문에, 동시에 여러 클라이언트들을 지원하는 비디오 서버는 디스크와 함께 많은 양의 메모리를 요구한다.

디스크 배열에 비디오 데이터를 저장하는 방법은 동

시에 처리할 수 있는 최대 디스크 I/O의 양과 밀접한 관계가 있다. 대부분의 비디오 서버는 디스크 배열에 속하는 디스크들 사이에 부하 분산을 위하여 하나의 비디오 데이터를 분할하여 여러 개의 디스크에 분산 저장하는 스트라이핑(striping) 기법을 사용한다[1][11][12][13][14]. 이러한 스트라이핑 기법은 병렬 입출력을 가능하게 하여 대용량의 데이터를 읽고 쓰는 응용에서 높은 전송률을 제공할 수 있다[15].

스트라이핑은 데이터 분할 단위에 따라 크게 세밀 스트라이핑 방법(fine-grained striping)과 굵은 스트라이핑 방법(coarse-grained striping)으로 나누어 진다[12]. 세밀 스트라이핑 방법은 한번에 읽을 비디오 데이터  $S$ 를  $D$ 개의 디스크에  $S/D$ 의 크기로 분리하여 저장한다. 그리고 해당 비디오 데이터에 대한 읽기 요청이 들어오면,  $D$ 개의 디스크는 각각  $S/D$  크기의 비디오 데이터를 읽어서 전달한다. 이 방법에서는 모든 디스크가 하나의 요구를 처리하기 위해 병렬로 동작하기 때문에 디스크 읽기의 응답 시간이 매우 작다. 한편, 굵은 스트라이핑 방법은 비디오 데이터를 한번에 읽을 데이터의 크기  $S$ 의 단위로  $D$ 개의 디스크에 분리하여 저장한다. 이 방법에서 한번의 읽기 요청은 하나의 디스크가 처리하기 때문에,  $D$ 개의 디스크는 동시에  $S * D$ 양의 디스크 읽기를 처리할 수 있다. 그림 1은 스트라이핑의 두 가지 방법에 대하여 설명하고 있다. A1~A6는 한번의 디스크 읽기를 통해 읽어 들이는 데이터 크기가  $S$ 이다. 세 개의 디스크를 사용할 때, 그림 1-(a)에서는 A1~A6가 A1-1~A6-6의 작은 단위( $S/D$ )로 나뉘어 스트라이핑 되어 있고, 그림 1-(b)에서는 A1~A6가  $S$  크기 단위로 스트라이핑 되어 있다. 일반적으로 굵은 스트라이핑 방법이 세밀 스트라이핑 방법보다 디스크의 처리 효율면에서 더 우수하다고 알려져 있기 때문에[12][13], 본 논문에서 고려하는 비디오 서버의 디스크 배열은 굵은 스트라이핑 방법을 사용한다고 가정한다. 굵은 스트라이핑 방법을 사용할 때,  $D$ 개의 디스크를 가진 디스크 배열은 하나의 디스크보다 처리량이  $D$ 배가 되며, 메모리의 사용량도 하나의 디스크에서 서비스할 때 필요한 양의  $D$ 배가 된다[8].



(a) fine-grained striping      (b) coarse-grained striping

그림 1 디스크 배열에서의 데이터 스트라이핑 방법

비디오 서버에서 메모리는 디스크들로부터 읽어 들인 비디오 데이터를 저장한 후 네트워크를 통하여 내보내는 일종의 버퍼링(buffering) 역할을 한다. 초기의 비디오 서버는 버퍼링을 위해 비디오 스트림마다 독립적인 메모리를 할당하여 사용하였다. 즉, 비디오 서버는 디스크로부터 비디오 스트림마다 할당된 버퍼 메모리에 비디오 데이터를 전달하고, 이 데이터를 한 주기 동안 네트워크를 통하여 지속적으로 전송한다. 그리고 다음 주기에는 다음에 이어질 비디오 데이터를 읽고 다시 전송한다. 하지만 이 방법은 이미 네트워크를 통하여 전송된 비디오 데이터도 계속 메모리 공간을 차지하고 있기 때문에 메모리의 사용이 매우 비효율적이다. 메모리의 사용량을 줄이기 위한 노력으로 버퍼 메모리를 풀(pool)의 형태로 만들어서 공유하여 사용하는 방법이 제안되었다[1][6]. 이 방법은 네트워크를 통하여 이미 전송된 비디오 데이터가 점유하고 있던 메모리 공간을 다음 비디오 데이터의 버퍼링에 사용한다. 이러한 방식으로 버퍼 메모리를 공유하면 필요한 메모리 양이 약 절반으로 줄어든다[6]. 따라서, 본 논문에서 고려하는 비디오 서버는 [1]과 [6]에서 제안한 버퍼 메모리 공유 방법을 사용한다고 가정한다.

그러므로, 본 논문에서는 그림 2와 같이 굵은 스트라이핑 방법을 지원하는 디스크 배열과 풀 형식의 버퍼 메모리 공유 방법을 지원하는 메모리로 구성된 비디오 서버를 고려한다.

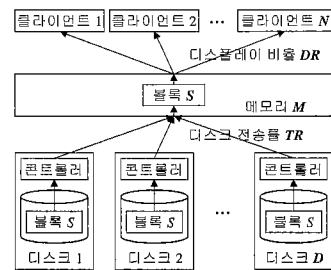


그림 2 비디오 서버 모델

### 3. VBR 비디오 서버의 자원 구성

본 논문에서 제안하는 확률 모델 기반의 VBR 비디오 서버의 자원 구성은 개략적으로 그림 3과 같은 입출력 값을 가진다. 이 방법은 서버의 요구 사항, 디스크 정보, 메모리 정보, 그리고 VBR 비디오 정보를 바탕으로 하드웨어 비용을 최소화시키는 디스크 수( $D$ ), 메모리 양

( $M$ ), 그리고 최적 디스크 접근 주기( $T_{cycle}^*$ )를 결정하는 것이다. 표 1은 그림 3에서 언급된 입출력 변수들에 관한 설명이다.

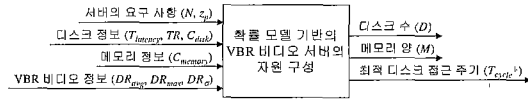


그림 3 VBR 비디오 서버의 자원 구성

표 1 비디오 서버 자원 구성의 입출력 변수

변수	설명	
입력 변수	$N$	동시에 서비스 가능한 클라이언트 수 또는 요구(request) 수의 최대 값으로 서버의 처리 능력을 나타낸다.
	$Z_p$	표준 정규 분포의 $p$ 100 (%) 백분위수 값이다. 여기서 $p$ 의 값은 비디오 서비스의 실시간 제약 조건이 보장될 확률이다.
	$T_{latency}$	디스크의 접근 지연 시간이다. 지연 시간은 탐색(seek) 지연 시간과 회전(rotation) 지연 시간으로 나눌 수 있다.
	$TR$	디스크 전송률을 나타낸다.
	$C_{disk}$	디스크 하나의 가격을 나타낸다.
	$C_{memory}$	메모리 1MB 당 가격을 나타낸다.
	$DR$	VBR 비디오의 디스플레이 비율로서 시간에 따라 상이한 값을 가진다. $DR_{avg}$ 는 $DR$ 확률 분포에서의 평균값을, $DR_{max}$ 는 최대값을, $DR$ 는 표준 편차를 나타낸다.
출력 변수	$D$	디스크의 수를 나타낸다.
	$M$	메모리의 양을 나타낸다.
	$T_{cycle}^*$	디스크와 메모리의 하드웨어 비용을 최소화시키는 비디오 서버의 디스크 접근 주기이다.

하나의 디스크에서 보다 많은 비디오 스트림을 처리하려면, 한번에 읽는 비디오 데이터 양을 늘려야 하지만 반대로 메모리 요구량이 급격히 늘어난다. 따라서 디스크와 메모리의 사용이 비용상 균형을 이루도록 디스크 하나가 처리해야 할 비디오 스트림 수를 찾는 것이 중요하다. 여기서 비용상 균형을 이룬다는 것은 비디오 스트림 하나를 처리하는데 필요한 디스크와 메모리의 경제적 비용을 최소로 한다는 의미이다.

비디오 서버가 처리하고자 하는 비디오 스트림 수가 결정되면, 요구되는 디스크 수는 디스크 하나가 처리할 비디오 스트림 수에 의해서 계산이 되고 메모리 양은 그 때 필요한 디스크 수와 디스크 당 요구되는 메모리 양에 따라 결정된다.

본 논문에서 제안하는 비디오 서버 자원 구성 방법은 먼저 그림 3에서 언급한 입력 정보로부터 디스크 접근 주기( $T_{cycle}$ ), 디스크 당 메모리 요구량( $M_{per-disk}$ ), 비디오 스트림 당 비용( $C_{per-stream}$ ), 디스크 당 최적 비디오 스트림 수( $N_{per-disk}^*$ ) 등의 중간 결과 값을 계산한다. 이후 이들 중간 결과들을 이용하여 최종적으로 디스크 배열에서의 디스크 수( $D$ ), 메모리 양( $M$ ), 그리고 최적 디스크 접근 주기( $T_{cycle}^*$ )를 결정한다.

여기서 우리는 그림 3의 입력 값들 중에서 VBR 비디오의 디스플레이 비율( $DR$ ) 정보를 결정하는 방법에 대해서 고려할 필요가 있다. VBR 비디오에서  $DR$ 은 시간에 따라 변하는 함수이므로  $DR$ 의 특성을 보다 잘 반영하기 위해 평균값( $DR_{avg}$ )과 최대값( $DR_{max}$ ) 같은 상수 값을 사용하기 보다는  $DR$  확률 분포 함수를 이용하는 것이 바람직하다. 아래 3.1절에서는 확률 모델을 기반으로  $DR$ 값을 결정하는 방법에 대해서 언급한다. 그리고 3.2절에서는 위에서 언급한 네 가지 중간 변수 값들을 계산하고, 3.3절에서는 3.2절의 중간 결과를 이용하여 그림 3에서 언급한 것과 같이 세 가지 최종 결과 값들에 대해서 설명한다.

### 3.1 확률 모델 기반의 VBR 비디오의 디스플레이 비율( $DR$ )

VBR 비디오의 디스플레이 비율( $DR$ )은 그림 4의 예와 같이 시간 함수로서 나타난다. 이러한 디스플레이 비율은 여러 개의 샘플 MPEG-1 비디오 데이터를 이용하여 확률 분포를 분석한 결과 정규 분포와 근사함을 알 수 있었다. 그림 5는 예제 비디오 데이터를 분석하여 얻은 디스플레이 비율의 확률 분포 중의 하나이다. 본 논문에서는 이러한 사실을 바탕으로 다음과 같은 가정을 한다. 그림 11의 실험은 가정의 타당함을 간접적으로 보인다.

가정 3.1 VBR 비디오의 디스플레이 비율은 정규 분포를 따른다고 가정한다.

가정 3.2 한 비디오 서버가 지원하는 VBR 비디오들은 모두 동일한 정규 분포를 가진다.

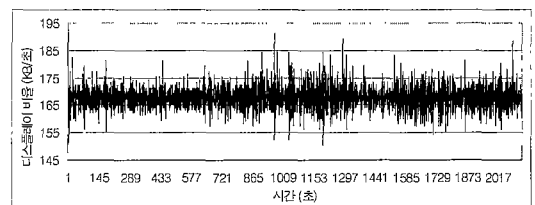


그림 4 시간에 따라 변하는 디스플레이 비율

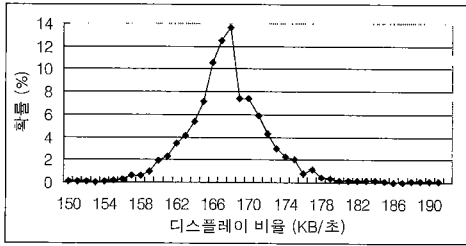


그림 5 디스플레이 비율의 확률 분포

따라서, 디스플레이 비율(DR) 값으로 DR 시간 함수의 평균값( $DR_{avg}$ )을 사용하면 확률적으로 비디오 서비스 기간의 반 가량은 디스크와 메모리 자원이 부족하여 비디오 스트림 서비스의 실시간 제약 조건을 만족하지 못하게 되고, 최대값( $DR_{max}$ )을 사용한다면 대부분의 서비스 기간 동안 디스크와 메모리 자원을 낭비하게 된다. 그러므로, 본 논문에서는 확률 모델을 기반으로 디스플레이 비율(DR)의 값을 결정하고자 한다. 아래 3.2절과 3.3절에서, 실제로 사용되는 디스플레이 비율(DR) 값은 서로 다른 VBR 비디오들의 디스플레이 비율 합인  $\sum_{i=1}^{N_{per-disk}} DR_i$ 이므로, 여기에서는 비디오 서버의 실시간 제약 조건을 확률  $p$ 만큼 보장하는 비디오 서버 자원을 구성하기 위한  $\sum_{i=1}^{N_{per-disk}} DR_i$  값을 확률 모델을 이용해서 구하고자 한다. 이때,  $N_{per-disk}$ 는 디스크 당 처리할 비디오 스트림 수를 의미한다.

<DR의 확률분포>

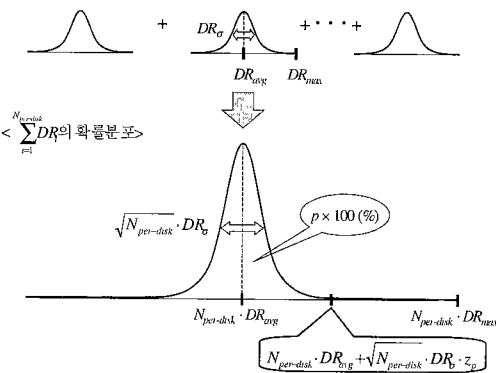


그림 6  $\sum_{i=1}^{N_{per-disk}} DR_i$ 의 확률 분포

위의 가정을 바탕으로 그림 6에서와 같이 임의의

VBR 비디오  $i$ 의 디스플레이 비율  $DR_i$ 는 평균값  $DR_{avg}$ , 최대값  $DR_{max}$ , 표준편차  $DR_{\sigma}$ 를 가지는 정규 분포를 따를 때,  $N_{per-disk}$  개의 VBR 비디오의 디스플레이 비율 합  $\sum_{i=1}^{N_{per-disk}} DR_i$ 는 평균값  $N_{per-disk} \cdot DR_{avg}$ , 최대값  $N_{per-disk} \cdot DR_{max}$ , 표준 편차가  $\sqrt{N_{per-disk}} \cdot DR_{\sigma}$ 인 정규 분포를 따른다. 정규 분포에서 표준 편차의 값은 그래프의 폭을 나타내며, 그래프의 폭이 좁을수록 디스플레이 비율이 평균값에 가까운 값을 가질 확률이 높다. 그러므로, 실시간 제약 조건을 확률  $p$ 만큼 보장할 수 있는 비디오 서버 자원 구성의 기준이 되는 디스플레이 비율 합( $\sum_{i=1}^{N_{per-disk}} DR_i$ )의 값은

$$\sum_{i=1}^{N_{per-disk}} DR_i = N_{per-disk} \cdot DR_{avg} + \sqrt{N_{per-disk}} \cdot DR_{\sigma} \cdot z_p \quad (1)$$

가 된다. 여기서  $z_p$ 는 표준 정규 분포의  $p \times 100(\%)$  백분위수 값이다. 추가적으로, 비디오 서버에서 실시간 제약 조건 위반을 허용하는 정도  $(1-p)$ 에 따른  $z_p$ 의 값은 표 2와 같다.

표 2 실시간 제약 조건 위반의 허용 정도(1-p)와  $z_p$ 의 값

허용 정도(1-p)	p	$z_p$
하루 동안 1초 (0.000011574)	0.999988426	4.3
한달 동안 1초 (0.000000386)	0.999999614	5.1
일년 동안 1초 (0.000000032)	0.999999968	5.5
백년 동안 1초 (0.00000000032)	0.9999999968	6.2
허용 없음 (0)	1	$\infty$

### 3.2 비디오 서버 자원을 위한 중간 변수 값

[디스크 접근 주기( $T_{cycle}$ )]

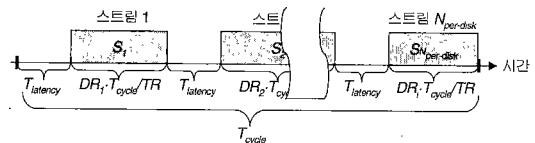


그림 7 디스크 접근 주기 ( $T_{cycle}$ )

비디오 서버는 그림 7과 같이 디스크 접근 주기  $T_{cycle}$  동안 디스크 당 처리할 비디오 스트림 수( $N_{per-disk}$ )만큼의 비디오 스트림에 대하여 한번 씩 디스크 읽기를 요청한다. 비디오 데이터  $i$ 의 디스플레이 비율을  $DR_i$ 라 할 때, 한번의 디스크 읽기에서 비디오 데이터  $i$ 에 대해 읽어 들이는 데이터 크기  $S_i$ 는 최소한 디스크 접근 주기

동안 디스플레이 할 수 있는 크기가 되어야 하므로  $S_i$ 는  $DR_i \cdot T_{cycle}$ 가 된다. 그리고,  $S_i$ 를 전송하는데 걸리는 시간  $T_{transfer, i}$ 는  $S_i/TR = DR_i \cdot T_{cycle} / TR$ 이 된다. 그러므로, 하나의 비디오 스트림  $i$ 를 디스크로부터 읽어 들이는 총 소요 시간  $T_{I/O, i}$ 는  $T_{latency} + T_{transfer, i} = T_{latency} + DR_i \cdot T_{cycle} / TR$ 이 된다. 결국, 디스크 접근 주기 동안  $N_{per-disk}$ 개의 비디오 스트림을 디스크로부터 읽어 들여야 하므로 디스크 접근 주기  $T_{cycle}$ 은 다음과 같다.

$$T_{cycle} = \sum_{i=1}^{N_{per-disk}} T_{I/O, i} = \sum_{i=1}^{N_{per-disk}} \left( T_{latency} + \frac{DR_i \cdot T_{cycle}}{TR} \right)$$

$$T_{cycle} = N_{per-disk} \cdot T_{latency} \cdot TR / \left( TR - \sum_{i=1}^{N_{per-disk}} DR_i \right) \quad (2)$$

여기서, 3.1절에서 언급한 식 (1)을 이용하면 최종적으로 디스크 접근 주기  $T_{cycle}$ 은 다음과 같이 계산된다.

$$T_{cycle} = \frac{N_{per-disk} \cdot T_{latency} \cdot TR}{TR - (N_{per-disk} \cdot DR_{avg} + \sqrt{N_{per-disk}} \cdot DR_{\sigma} \cdot Z_p)} \quad (3)$$

**[디스크 당 메모리 요구량( $M_{per-disk}$ )]**

디스크 당 메모리 요구량( $M_{per-disk}$ )을 구하기 위해서는 먼저 각 비디오 스트림의 메모리 요구량을 알아야 한다. 그림 8과 같이 한 비디오 스트림은 디스크 읽기를 요청한 시점부터 끝나는 시점까지 읽어 들일 데이터의 양  $S$ 만큼의 메모리를 요구한다. 그리고 디스크 읽기가 끝나면 다음 디스크 접근 주기까지 메모리 요구량은  $DR$ 의 비율로 감소한다.

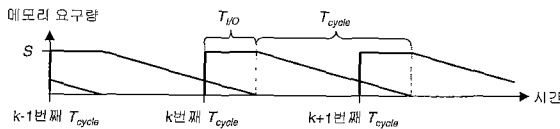
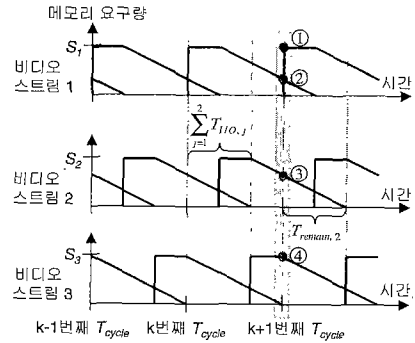
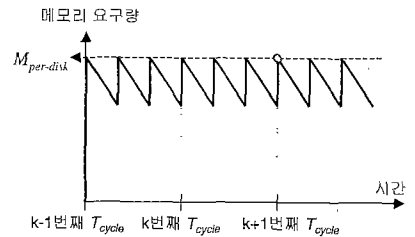


그림 8 비디오 스트림 당 메모리 요구량

만약 디스크 하나가 3개의 비디오 스트림을 처리한다고 할 때, 디스크 접근 주기  $T_{cycle}$  동안 3개의 비디오 스트림을 비디오 스트림 1, 2, 3의 순서로 디스크로부터 읽는다. 이들 비디오 스트림의 메모리 요구량은 그림 9(a)와 같으며, 디스크 당 메모리 요구량( $M_{per-disk}$ )은 그림 9(b)와 같이 각 비디오 스트림의 메모리 요구량을 더한 것이 된다. 여기서,  $T_{I/O, i}$ 는 비디오 스트림  $i$ 를 디스크로부터 읽어 들이는 총 소요시간이며,  $T_{remain, i}$ 는 디스크 접근 주기 시작 지점에서 이전 디스크 접근 주기에서 읽은 비디오 스트림  $i$ 를 모두 전송하는데 소요되는 시간을 의미한다.



(a)



(b)

그림 9 디스크 당 메모리 요구량

그림 9(b)에서 디스크 당 메모리 요구량  $M_{per-disk}$  값을 구하기 위해서  $k+1$ 번째  $T_{cycle}$  지점에서의  $M_{per-disk}$  값을 고려하자. 결국,  $k+1$ 번째  $T_{cycle}$ 에 표시된 그림자 영역 안에 표시된 네 개의 작은 원들에서 요구하는 메모리의 양의 합이  $M_{per-disk}$  값이 된다. 첫 번째 원은 비디오 스트림 1에 대해서 다음 디스크 접근 주기 동안 디스크로부터  $S_1$ 만큼의 데이터를 메모리로 읽어 들이므로  $S_1$  크기의 메모리가 필요함을 의미하고, 두 번째 원은 이전 디스크 접근 주기에서 읽은 비디오 스트림 1중에서 전송되지 않고 남은 데이터에 대해 메모리가 필요함을 의미한다. 세 번째와 네 번째 원은 각각 비디오 스트림 2와 3의 전송되지 않고 남은 데이터가 메모리를 점유하고 있음을 나타낸다. 여기서, 그림 9(a)에서의 두 번째, 세 번째, 네 번째 원이 의미하는 비디오 스트림  $i$ 의 전송되지 않고 남은 데이터가 메모리에 있는 양은 각각  $S_i \cdot T_{remain, i} / T_{cycle}$ 이 된다. 그리고,  $T_{remain, i}$ 의 시간은 그림 9(a)에서 보통이 비디오 스트림 1에서부터 비디오 스트림  $i$ 까지 디스크에서 읽어 들이는 시간인  $\sum_{j=1}^i T_{I/O, j}$ 와 동일하다. 현재 디스크 접근 주기에서 읽어 들이는 비디오 스트림  $i$ 의 양을  $S_i'$ 이라고 하고 이전 디스크 접근 주기에서 읽은 비디오 스트림  $i$ 의 양을  $S_i$ 라

할 때,  $M_{per-disk}$ 는 다음과 같다.

$$M_{per-disk} = \text{그림 9(a)의 ②, ③, ④에 대한 메모리 요구량} + \text{①에 대한 메모리 요구량}$$

$$= \sum_{i=1}^{N_{per-disk}} S_i \cdot \sum_{j=1}^i T_{1/O,j} / T_{cycle} + S_1'$$

$$= \sum_{i=1}^{N_{per-disk}} DR_i \cdot \sum_{j=1}^i \left( T_{latency} + \frac{DR_j \cdot T_{cycle}}{TR} \right) + DR_1' \cdot T_{cyc} \quad (4)$$

여기서  $DR_i'$  값은 디스크 접근 주기에 따라 변하는 시간 함수이므로 모든 경우를 포함할 수 있는  $DR_{max}$  값을 사용하기로 하고,  $\sum_{i=1}^{N_{per-disk}} i \cdot DR_i$ 와  $\sum_{i=1}^{N_{per-disk}} \sqrt{i} \cdot DR_i$ 의 값은 3.1절에 언급한 확률 모델을 기반으로 하면 다음과 같다.

$$\sum_{i=1}^{N_{per-disk}} i \cdot DR_i(t_i) = \frac{N_{per-disk} \cdot (N_{per-disk} + 1)}{2} \cdot DR_{avg}$$

$$+ \sqrt{\frac{N_{per-disk} \cdot (N_{per-disk} + 1)}{2} \cdot (2 \cdot N_{per-disk} + 1)} \cdot DR_{\sigma} \cdot z_p$$

$$\sum_{i=1}^{N_{per-disk}} \sqrt{i} \cdot DR_i(t_i) = \sum_{i=1}^{N_{per-disk}} \sqrt{i} \cdot DR_{avg}$$

$$+ \sqrt{\frac{N_{per-disk} \cdot (N_{per-disk} + 1)}{2}} \cdot DR_{\sigma} \cdot z_p$$

위의 결과들과 식 (3)의  $T_{cycle}$  값을 통하여 최종적으로 디스크 당 메모리 요구량  $M_{per-disk}$ 는 다음과 같이 표현된다.

$$M_{per-disk} = \left( T_{latency} + \frac{N_{per-disk} \cdot T_{latency} \cdot DR_{avg}}{TR - (N_{per-disk} \cdot DR_{avg} + \sqrt{N_{per-disk} \cdot DR_{\sigma} \cdot z_p})} \right) \cdot \left( \frac{N_{per-disk} \cdot (N_{per-disk} + 1)}{2} \cdot DR_{avg} + \sqrt{\frac{N_{per-disk} \cdot (N_{per-disk} + 1)}{2} \cdot (2 \cdot N_{per-disk} + 1)} \cdot DR_{\sigma} \cdot z_p \right) + \frac{N_{per-disk} \cdot T_{latency} \cdot DR_{\sigma} \cdot z_p}{TR - (N_{per-disk} \cdot DR_{avg} + \sqrt{N_{per-disk} \cdot DR_{\sigma} \cdot z_p})} + \left( \sum_{i=1}^{N_{per-disk}} \sqrt{i} \cdot DR_{avg} + \sqrt{\frac{N_{per-disk} \cdot (N_{per-disk} + 1)}{2}} \cdot DR_{\sigma} \cdot z_p \right) + \frac{N_{per-disk} \cdot T_{latency} \cdot TR \cdot DR_{max}}{TR - (N_{per-disk} \cdot DR_{avg} + \sqrt{N_{per-disk} \cdot DR_{\sigma} \cdot z_p})}$$

**[비디오 스트림 당 비용( $C_{per-stream}$ )]**

비디오 스트림 하나에 대한 하드웨어 비용은 디스크, 메모리, 중앙 처리 장치(CPU), 주변 기기 등 여러 하드웨어의 비용을 포함하고 있다. 그러나 일반적으로 주변 기기들은 비디오 스트림 수와 무관하므로 본 논문에서는 디스크와 메모리의 비용만을 고려하기로 한다. 따라서 하나의 디스크가  $N_{per-disk}$ 개수의 비디오 스트림을 지원한다고 할 때, 비디오 스트림 하나 당 비용( $C_{per-stream}$ )은 비디오 스트림 하나를 처리하기 위해 필요한 디스크와 메모리 비용으로 다음과 같다.

$$C_{per-stream} = \frac{C_{disk} + C_{memory} \cdot M_{per-disk}}{N_{per-disk}} \quad (6)$$

**[디스크 당 최적 비디오 스트림 수( $N_{per-disk}^*$ )]**

디스크 하나가 처리할 최적의 비디오 스트림 수 ( $N_{per-disk}^*$ )는 비디오 스트림 당 비용  $C_{per-stream}$ 을 최소로 하는 디스크 당 비디오 스트림 수가 된다. 비디오 스트림 당 비용에서 디스크 비용은 디스크 당 비디오 스트림 수에 따라 그림 10과 같은 함수를 가진다[9]. 디스크 당 비디오 스트림 수  $N_{per-disk}$ 는 1보다 크거나 같고  $\sum_{i=1}^{N_{per-disk}} DR_i < TR$ 를 만족시키는 정수이므로, 디스크당 최적 비디오 스트림 수  $N_{per-disk}^*$ 는 이 범위에서 비디오 스트림 당 비용  $C_{per-stream}$ 을 최소로 하는 정수 값이 된다.

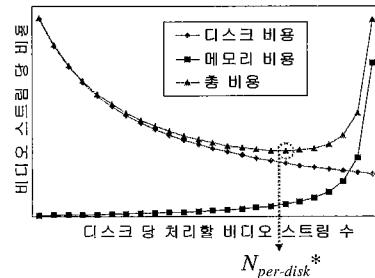


그림 10 비디오 스트림 당 비용

**3.3 확률 모델 기반의 최적 비디오 서버 자원의 구성**

여기에서는 3.1과 3.2절의 결과를 바탕으로 본 논문에서 제안하는 확률 모델 기반의 최적 자원 구성 방법의 최종 결과인 디스크 수(D), 메모리 양(M), 최적 디스크 접근 주기( $T_{cycle}^*$ )를 결정하고자 한다. 본 논문에서 고려하는 비디오 서버 모델에서 디스크 하나가 처리해야 하는 최적 비디오 스트림의 수를  $N_{per-disk}^*$ 로 유지하면서 디스크 배열을 사용하는 것이 전체적으로 최적의 상태를 유지할 수 있다. 왜냐하면 2장에서 언급했듯이 스트라이핑 단위가 큰 방법을 사용하는 비디오 서버 모델에서는 D개의 디스크로 구성된 배열은 디스크 하나보다 처리량이 D배이며, 메모리 사용량 역시 D배이기 때문이다. 그러므로 비디오 서버가 동시에 처리하고자 하는 비디오 스트림 수, 즉 서비스할 클라이언트 수를 N이라고 할 때, 비디오 서버 구성에 필요한 최적 디스크 수 D는 다음과 같다.

$$D = \left\lceil \frac{N}{N_{per-disk}^*} \right\rceil \quad (7)$$

그리고, 비디오 서버에서 필요한 최적의 메모리 양  $M$ 은 디스크 당 최적 메모리 요구량( $M_{per-disk}^*$ )의  $D$ 배이므로

$$M = D \cdot M_{per-disk}^* = \left[ \frac{N}{N_{per-disk}^*} \right] \cdot M_{per-disk}^* \quad (8)$$

이다. 여기서,  $M_{per-disk}^*$ 는 3.2절의 식 (5)에서  $N_{per-disk}$  대신에  $N_{per-disk}^*$  값을 대입한 식이다. 위에서 우리는 식 (7), (8)을 통해 처리하고자 하는 비디오 스트림 수가 주어졌을 때, 최적의 디스크 수와 메모리 양을 결정하는 방법에 대하여 알아보았다. 이 때 고려해야 할 사항은 디스크와 메모리를 최적으로 구성하였다 하더라도 이를 효율적으로 사용하지 않는다면 원하는 처리량을 얻지 못하기 때문에, 디스크 접근 주기(즉, 한번의 디스크 읽기에서 얻는 데이터의 상영 시간)를 최적화해야 한다. 최적 디스크 접근 주기  $T_{cycle}^*$ 는 3.2절의 식 (3)의  $T_{cycle}$ 에서  $N_{per-disk}$  대신에  $N_{per-disk}^*$ 를 대입한 식이 된다.

#### 4. 실험

비디오 스트림의 디스플레이 비율( $DR$ )의 값은 본 논문에서 제안한 확률 모델 이외에 평균값, 최대값 등과 같이 하나의 통일된 상수 값을 이용하여 결정할 수 있다. 본 실험에서는 확률 모델을 이용한  $DR$  값과 평균값 및 최대값을 이용한  $DR$  값의 유용성을 비교 평가하고자 한다.

본 실험은 Sun Sparc 2 CPU 환경에서 해상도 352×240의 MPEG-1 비디오 데이터를 이용하여 2592000초(한 달) 동안의 비디오 서비스를 시뮬레이션하여  $DR$  합의 변화를 측정한 결과이다. 여기서, 비디오 데이터의  $DR_{avg}$ 는 168KB/초,  $DR_{max}$ 는 191KB/초,  $DR_{min}$ 는 150KB/초,  $DR$ 는 4.36KB/초이다.

3장에서  $DR$  값은 식 (2)의 디스크 접근 주기( $T_{cycle}$ )와 식 (4)의 디스크 당 메모리 요구량( $M_{per-disk}$ )을 계산하는데 사용된다. 실제로 이들 식에서는 VBR 비디오들의 디스플레이 비율 합인  $\sum_{i=1}^N DR_i$ 의 형태로 사용되므로 이 값이 각각 확률 모델, 평균값, 최대값을 이용하여 계산될 때를 서로 비교하고자 한다.  $DR$ 의 평균값을 사용하면,  $\sum_{i=1}^{N_{per-disk}} DR_i$ 는  $N_{per-disk} \cdot DR_{avg}$ 의 값을 가진다.  $DR$ 의 최대값을 사용하면  $\sum_{i=1}^{N_{per-disk}} DR_i$ 는  $N_{per-disk} \cdot DR_{max}$ 의 값을 가진다. 그리고, 3.1절에서 언급했듯이  $DR$ 의 확률 모델을 이용하면,  $\sum_{i=1}^{N_{per-disk}} DR_i$ 는  $N_{per-disk} \cdot DR_{avg} \sqrt{N_{per-disk}} \cdot DR_{\sigma} \cdot z_p$ 의 값을 가진다.

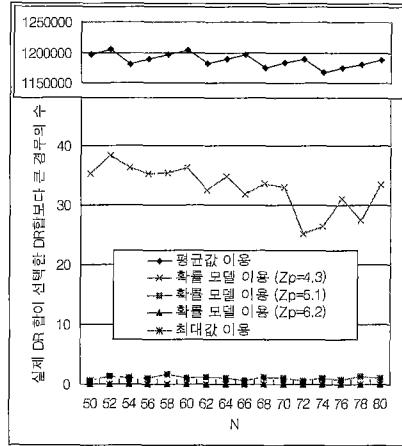


그림 11 비디오 서버의 실시간 제약 조건의 만족 비율

그림 11은 디스크 하나가 서비스할 수 있는 비디오 스트림의 수  $N_{per-disk}$ 의 값을 50개에서 80개까지 변화시킬 때 2592000초(한달) 동안 실제 비디오  $DR$  합이 위에서 언급한 확률 모델, 평균값, 그리고 최대값을 이용하여 계산된  $DR$  합을 초과한 시간을 나타낸다. 확률 모델을 이용한  $DR$  합의 계산에서  $z_p$ 의 값을 4.3, 5.1, 그리고 6.2의 세가지 경우로 나누었다. 여기서, 실제 비디오  $DR$  합이 계산을 통하여 선택된  $DR$  합을 초과한다는 것은 초과된 시간 동안 비디오 서버는 동시에 여러 비디오 스트림을 상영하는데 요구되는 실시간 제약 조건을 만족하지 못함을 의미한다. 그림 11에서와 같이 평균값을 이용하여  $DR$  합을 계산한 경우 비디오 서비스 시간의 약 50% 동안 실시간 제약 조건을 만족하지 못한다. 반면, 최대값을 사용한 경우 예상되는 것처럼 항상 비디오 서버의 실시간 제약 조건을 만족함을 쉽게 알 수 있다. 확률 모델을 이용하여  $DR$  합을 계산한 경우  $z_p$ 의 값에 따라 차이가 있지만 실시간 제약 조건을 어기는 경우가 충분히 받아들여질 수 있을 만큼 상당히 낮음을 관찰할 수 있다.

그림 11의 결과로 미루어보아 최대값을 사용하여 비디오  $DR$  합을 계산하면 여러 비디오 스트림을 상영하는데 요구되는 비디오 서버의 실시간 제약 조건을 항상 만족할 수 있지만, 그림 12의 실험 결과는 최대값을 사용하면 많은 비디오 서버의 자원을 낭비하게 됨을 보여준다. 그림 12는 디스크 하나가 서비스할 수 있는 비디오 스트림의 수  $N_{per-disk}$ 에 따라 본 논문에서 계산된  $DR$  합과 실제 실험에서 나타나는 최대  $DR$  합의 비율을 나타낸다. 이 비율 값이 1에 가까울수록 계산된  $DR$



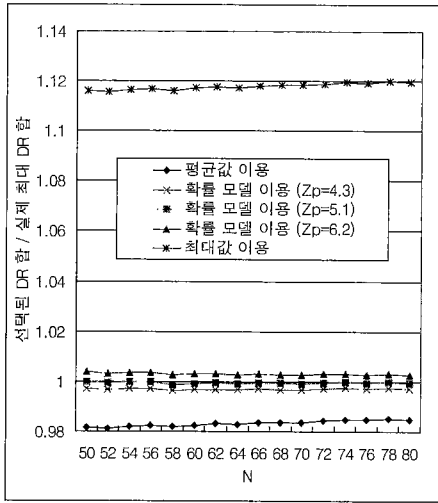


그림 12 비디오 서버의 자원 활용 비율

합을 이용하면 더욱 효율적으로 비디오 서버의 자원 구성을 할 수 있음을 나타낸다. 최대값을 이용하여 계산된 DR 합인 경우 이 비율의 값이 1보다 높게 나타나며, 이는 실제 비디오 서버를 운용할 때 최대값을 이용하여 계산된 DR 합만큼의 자원이 거의 필요하지 않음을 나타낸다.

그러므로, 그림 11과 12의 실험 결과를 통하여 본 논문에서 제안하는 확률 모델 기반의 디스플레이 비율을 결정하는 것이 가장 유용함을 알 수 있다.

## 5. 결론

기존의 연구된 CBR 자원 구성 방법을 이용하여 VBR 비디오 서버의 자원을 구성하기 위하여 VBR 비디오 디스플레이 비율의 평균값이나 최대값을 사용할 경우, 비디오 스트림 서비스의 실시간 제약 조건을 자주 만족하지 못하거나 많은 자원을 낭비하게 되는 문제점이 있음을 관찰할 수 있었다. 본 논문에서는 이러한 문제를 해결하기 위해서 VBR 비디오 서비스에 요구되는 자원의 양을 정확하게 추정할 수 있는 확률 모델을 이용하여 VBR 서버의 자원을 구성하는 방법을 제안하였다. 이 방법은 비디오 서버의 처리량은 동일하게 유지하면서 디스크와 메모리의 하드웨어 비용을 최소화시킬 수 있는 비디오 서버의 접근 주기를 선택하고 이를 기반으로 디스크와 메모리의 양을 결정한다. 그리고, 실험을 통하여 확률 모델을 이용한 VBR 처리 방법은 평균 값 또는 최대값을 이용하는 방법보다 효율적이라는 것

을 알았다.

본 논문에서 제안한 자원 구성 방법은 MPEG이나 ASF 등을 서비스하는 비디오 서버의 효율적인 구축에 활용될 수 있을 것이다. 또한 논문에서 언급한 VBR 비디오 스트림의 자원 요구량에 대한 식은 앞으로 VBR 비디오 서버의 비용 모델로서 정의될 수 있다.

최근 서버와 클라이언트 사이의 네트워크 상태나 클라이언트 자체의 성능에 따라 다른 품질을 가진 비디오를 서비스하는 경우도 발생하고 있다. 이 경우 비디오 서버가 사용하는 비디오의 디스플레이 비율은 다른 확률 분포를 갖는다. 따라서 향후 디스플레이 비율이 서로 다른 확률 분포를 갖는 비디오를 서비스하는 비디오 서버의 자원 구성 방법에 대해 연구할 필요가 있다.

## 참고 문헌

- [1] Fouad A. Tobagi, Joseph Pang, Randall Baird, and Mark Gang, "Streaming RAID: A Disk Array Management System for Video Files," ACM Multimedia, pp.393-400, 1993.
- [2] A. Reddy and J. Wyllie, "I/O issues in a multimedia system," Computer 2, pp.69-74, 1994.
- [3] P. Yu, M.-S. Chen, and D. Kandlur, "Grouped sweeping scheduling for DASD-based multimedia storage management," Multimedia Systems, 1(1), pp.99-109, 1993.
- [4] S. Ghandeharizadeh, S. Kim, and C. Shahabi, "On configuring a single disk continuous media server," Sigmetrics Performance Evaluation, 23(1), pp.37-46, 1995.
- [5] Y.-J. Oyang and C.-H. Wen, "A multimedia storage system for on-demand playback," IEEE Transaction on Consumer Electronics, 41(1), pp. 53-64, 1995.
- [6] Edward Chang and Yi-Yin Chen, "Minimizing memory requirements in a multimedia storage system," Stanford Technical Report SIDL-WP-1996-0045, 1996.
- [7] R.Ng and J. Yang, "Maximizing buffer and disk utilizations for news on-demand," In Proceedings of the 20th VLDB Conference, pp.451-462, 1994.
- [8] Edward Chang and Hector Garcia-Molina, "Effective Memory Use in a Media Server," In Proceedings of the 23rd Very Large Data Base (VLDB) Conference, pp.496-505, 1997.
- [9] Edward Chang and Hector Garcia-Molina, "Cost-Based Media Server Design," In Proceedings of the 8th Research Issues in Data Engineering, pp. 76-83, 1998.
- [10] W. Shi and S. Ghandeharizadeh, "Trading memory

for disk bandwidth in video-on-demand servers," Computer science Technical Report 97-653, USC, 1997.

- [11] Steven Berson, Shahram Ghandeharizadeh, Richard R. Muntz, and Xiangyu Ju, "Staggered Striping in Multimedia Information Systems," SIGMOD Conference, pp.79-90, 1994.
- [12] Banu Ozden, Rajeev Rastogi, and Abraham Silberschatz, "Disk Striping in Video Server Environments," ICMCS, pp.580-589, 1996.
- [13] Kimberly Keeton and Randy H. Katz, "Evaluating Video Layout Strategies for a High-Performance Storage Server," Multimedia Systems 3(2), pp.43-52, 1995.
- [14] Prashant J. Shenoy and Harrick M. Vin, "Efficient Striping Techniques for Variable Bit Rate Continuous Media File Servers," In Proceeding of the Seventh IEEE International Workshop on Network and Operating System Support for Digital and Video (NOSSDAV), pp.25-36, 1997.
- [15] 김명호, 이윤준, 멀티미디어 개념 및 응용



이 윤 준

1977년 서울대학교 계산통계학과 졸업. 1979년 한국과학기술원 전산학과에서 석사학위 취득. 1983년 France, INPGEN-SIMAG에서 박사학위 취득. 1983년 ~ 1984년 France, IMAG 연구원. 1984년 ~ 현재 한국과학기술원 전산학과 교수. 1989년 MCC(미) 초빙연구원. 1990년 CRIN(불) 객원교수. 관심분야는 데이터베이스 시스템, 정보검색, 실시간 데이터베이스 등임.



조 대 현

1998년 경북대학교 전산학과 학사. 2000년 한국과학기술원 전산학과 석사. 2000년 ~ 현재 한국과학기술원 전산학과 박사과정 재학중. 관심분야는 지리 정보 시스템, 멀티미디어 서버, 웹 캐시, 스트리밍 미디어 캐시



손 진 현

1996년 서강대학교 전산학과 학사. 1998년 한국과학기술원 전산학과 석사. 2001년 한국과학기술원 전산학과 박사. 2001년 ~ 현재 한국과학기술원 전산학과 연구원. 관심 분야는 분산데이터베이스, 멀티데이터베이스, 미들웨어, 워크플로우, 데이터웨어하우징, 분산 스케줄링, CORBA.

김 명 호

정보과학회논문지 : 데이터베이스  
제 28 권 제 1 호 참조