지연시간의 변화를 고려한 연속미디어 표현 메카니즘의 설계 및 분석

여 인 국† 황 대 훈††

요 약

본 논문에서는 광디스크, 디지탈 테이프 등과 같은 제3의 저장 장치의 활용을 높이기 위하여 다중의 디스크와 CD-ROM으로 구성된 저장구조를 제안하고, 이 구조에서 CD-ROM에 저장된 연속미디어 객체들을 효율적으로 표현하는 메카니즘을 설계한다. 본 논문에서 제안한 연속미디어 저장 구조는 대역폭이 높은 SCSI 버스를 다중의 디스크가 공유하여 데이타를 전송함으로써 정보의 검색 능력을 향상시킨다. 그리고 연속미디어 표현 메카니즘은 CD-ROM의 연속미디어 객체의 인출과 표현을 병렬로 수행시켜 연속미디어 표현의 불연속을 감소시킬 뿐 아니라 자주 참조되는 객체를 디스크에 존재케 함으로써 재참조 요청에 대한 실기율과 서비스 시간의 변화를 최소화한다.

Design and Analysis of Continuous Media Representation Mechanism to minimize the Variance of Latency Time

In Kook Yeo† and Dae Hoon Hwang††

ABSTRACT

In this paper, a mechanism to enhance the utilization of the tertiary storage devices such as optical disk, digital tape and so on is proposed. For this purpose, we suggest a storage structure with a CD-ROM and multiple disk and design an efficient mechanism to represent the continuous media stored in CD-ROM. The continuous media storage structure proposed in this paper can enhanced the retrieval capability of information by sending data using a shared bus with high bandwidth. And a continuous media representation mechanism not only can reduce the discontinuity of representation using parallel operation of fetch and representation but also minimize the variance of service time and the missing ratio of re-reference requirement by residing frequently accessed object on the disk.

1. 서 론

정보기술의 발달과 함께 멀티미디어(multimedia) 정보처리 시스템의 보급이 확대되면서 오디오(audio), 비디오(video) 그리고 애니메이션(animation)과 같이 시간 지연에 민감한(delay sensitive) 연속미디어(continuous media)에 대한 실시간 처리가 요구되고 있으나 현재의 멀티미디어 정보 처리 시스템의 기술수준은 이를 충분히 지원하지 못하고 있다.

그 주요 원인으로는 연속미디어는 높은 대역폭(bandwidth)을 가지고 대형의 저장공간을 필요로 하는 반면에 현재의 통신매체나 기억장치의 입출력 대역폭은 상대적으로 낮기 때문이다. 특히 저장공간의 경우에 있어서는 자기디스크(magnetic disk)를 비롯하여 광디스크(optical disk)나 디지탈 테이프(digital tape)와 같은 제3의 저장장치(tertiary storage device)를 이용함으로써 용량에 대한 문제를 해결하고 있으나, 이들 장치의 낮은 데이타 전송율과 탐색지연(seek

† 정 회 원 : 생산기술연구원 선임연구원
†† 정 회 원 : 경원대학교 전자계산학과 부교수
논문접수 : 1995년 6월 9일, 심사완료 : 1995년10월2일

latency) 등으로 인하여 데이타의 표현에 대한 실시간 요구는 충분히 만족시키지 못하고 있는 실정이다.

따라서 이와 같은 문제를 해결하고자 연속미디어 저장 서버를 중심으로 대용량의 저장장치의 효율적 활용을 위한 연구가 활발히 진행되고 있다. 대표적인 연구방향으로는 디스크 시스템의 대역폭을 증가시키는 방법과 연속미디어 객체간의 동기화 및 실시간 재생을 위한 시간관계 해석 등의 연구가 이루어지고 있는데, 이들 대부분의 연구는 연속미디어 객체가 디스크에 존재하여 서비스를 실시하는 것을 전제로 하고있다.

그러나 현재의 멀티미디어 정보 제공 수단이 CD-ROM(Compact Disk Read Only Memory) 타이틀(title)이나 비디오 디스크, 그리고 디지탈 테이프 등과 같은 매체를 통하여 이루어지고 있는 점에 비추어 볼 때 저장 서버의 기능 향상을 위해서는 제3의 저장장치의 효율적 활용에 대한 연구가 함께 이루어져야 할 것이다.

따라서 본 논문에서는 연속미디어 객체가 디스크에 존재하지 않을 때 제3의 저장장치에 저장된 연속미디어를 효율적으로 서비스하기 위한 방법에 관해서 연구하였다. 그 중 제3의 저장장치에 저장된 다수의 연속미디어 객체들에 대해서 참조 요청이 자주 발생하는 경우에 이들을 연속적으로 표현할 수 있는 메카니즘을 설계하고 성능을 평가하였다.

본 논문의 구성은 다음과 같다. 먼저 제1장 서론에 이어 제2장에서는 연속미디어의 특징과 서비스 기술에 관련된 연구내용을 고찰한다. 제3장에서는 연속미디어의 실시간 표현을 위한 저장 구조를 제안하고, 제안된 저장 구조에서 연속미디어를 효율적으로 표현하기 위한 객체 인출 메카니즘과 디스크 공간 확보 알고리즘을 설계하며, 제4장에서 이에 대한 성능을 평가,분석한후 제5장에서 연구에 대한 결론을 맺는다.

2. 연속미디어의 특징 및 서비스 기술

2.1 연속미디어의 특징

연속미디어란 디지탈 오디오, 비디오, 애니메이션 등과 같이 정해진 시간 간격에 맞게 연속적으로 표현되는 미디어로서 높은 대역폭을 갖는 대형의 데이타로 구성된다.

예를들어 "Network-Quality" 비디오를 위해 NTSC(National Television System Committee)에서 요구하는 대역폭은 45mbps(mega bit per second) 정도이며, CCIR(International Radio Consultative Committee) 권고안 601은 216mbps를 요구하고 있다. 또한 HDTV에서의 비디오 객체에 대한 요구 대역폭은 800mbps이다. 이와 같이 높은 대역폭을 갖는 객체들의 크기를 계산하면 다음과 같은 결론을 얻을 수 있다. 즉, 압축되지 않은 30분 짜리 NTSC 비디오 클립(clip)은 대략 10GB(Giga Byte) 정도의 크기를 가지게 되는데, 이 객체에 대해서 1.5mbps로 대역폭을 감소시킨다고 하더라도 약 337MB(Mega Byte) 정도가 된다[1].

따라서 높은 대역폭과 대형의 크기를 갖는 연속미디어 객체들을 사용자나 시스템이 정의한 시간적인 제한 조건에 맞게 미디어 출력장치나 통신망을 통하여 서비스하기 위해서는 데이타 압축 기술을 사용하여 객체의 크기를 줄이거나, 저장장치의 데이타 처리율을 높이는 노력이 필요하다.

2.2 연속미디어 서비스 기술

연속미디어 정보를 사용자의 요구에 따라 실시간으로 서비스함에 있어서 반드시 고려되어야할 것이 미디어의 연속성과 미디어간의 동기화, 그리고 다중 사용자에 대한 동시 서비스인데, 이를 위해서 연속미디어 저장 서버는 대역폭이 큰 대용량의 저장 장치와 각 사용자 서비스의 실시간성 및 연속성을 제공하는 메카니즘을 제공해야 한다[2].

이에 연속미디어의 서비스에 관련된 기술 중 디스크의 성능을 향상시키기 위한 기존의 연구로는 다중의 디스크를 이용하여 탐색지연을 줄이고 전송율을 증가시키는 방법이 있다[3, 4]. 그리고 디스크 헤드의 스케쥴링을 통하여 디스크의 탐색율을 줄이는 방법과[5, 6], 미디어 객체의 검색 단위를 증가시킴으로써 단위시간당 디스크 탐색 횟수를 줄이는 방법이 있다[7, 8]. 이와함께 저속의 디스크로부터 수행될 데이타를 미리 버퍼로

읽어옴으로써 인출에 소요되는 지연을 줄이기 위한 선인출(prefetch) 전략 등이 있다[9, 10].

3. 연속미디어 표현 메카니즘의 설계

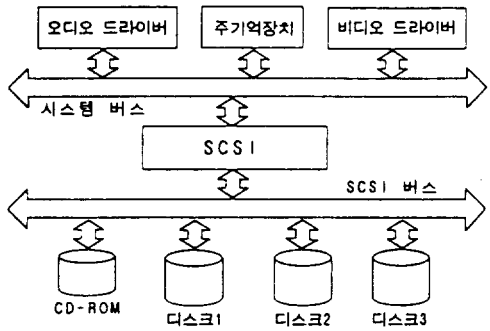
연속미디어의 표현에 있어서 서비스의 응답시간과 품질(quality)은 사용자의 참조 요청이 발생한 시점으로부터 실제 표현이 이루어지기까지의 시작 지연시간과 단일 객체의 데이타 블럭 또는 각 객체 사이의 표현의 연속성에 따라 결정된다. 이에 영화의 상영과 같이 연속미디어의 재생을 목적으로 하는 시스템에서는 시작 지연시간이 수행시간에 비하여 무시할 정도로 짧다. 그러나 멀티미디어 프리젠테이션 시스템이나 교육 훈련용 프로그램 등과 같이 다수의 연속미디어 객체에 대해서 참조 요청이 자주 발생하는 경우 각각의 표현에 대한 시작 지연시간과 불연속은 참조 요청 횟수와 객체의 크기에 비례하여 증가한다. 더우기 저장장치의 대역폭이 객체의 요구 대역폭보다 낮은 경우에는 시작 지연시간과 이로 인한 불연속은 더욱 커진다.

따라서 연속미디어의 실시간 서비스와 연속성 유지를 위해서는 높은 대역폭을 갖는 저장구조와 이를 지원하는 서비스 메카니즘의 설계가 요구된다.

3.1 연속미디어 저장 구조

본 논문에서 제안한 연속미디어 저장 구조는 주기억장치(main memory)와 다중의 디스크 그리고 CD-ROM으로 이어지는 저장 구조를 가지고 있다. 그리고 연속미디어를 표현하기 위한 출력장치로 오디오 드라이버(audio driver)와 비디오 드라이버(video driver)를 포함한다. 여기서 주기억장치는 디스크 또는 CD-ROM의 연속미디어 객체를 출력장치를 통하여 표현하는데 있어서의 선입선출(First-In First-Out) 형태의 버퍼로 이용된다. (그림 1)은 연속미디어 저장 구조를 나타낸 것으로 디스크와 CD-ROM 드라이브는 SCSI(Small Computer System Interface) 버스를 통하여 연결되어 있으며, 출력장치는 시스템 버스에 연결되어 있다. 이때 SCSI 버스와 시스템 버스는 SCSI를 통하여 연결된다. 〈표 1〉은 연속미디어 저장 구조를 구성하는 각 주변장치의 특성을 나타낸 것이다[5].

연속미디어 저장 구조의 동작은 개략적으로 다음과 같다. 연속미디어 객체에 대한 서비스 요청이 발생했을 때 시스템은 다중의 디스크에 분산되어 있는 연속미디어의 검색을 위해 각 디스크에 대해 라운드-로빈(round-robin) 방식으로 데이타의 전송 기회를 부여한다. 한편 참조 요청된 객체가 디스크에 존재하지 않을 경우에는 CD-ROM으로부터 직접 데이타의 읽기 동작을 수행하는데, 이 과정에서 시스템은 CD-ROM의 데이타를 출력장치로 표현함과 동시에 이들 데이타를 각 디스크에 디클러스터링(declustering)하여 저장한다. 이때 시스템은 자체의 버스를 통하여 SCSI와 명령 및 데이타를 주고받으며, CD-ROM 및 각 디스크는 SCSI의 명령에 따라 SCSI 버스를 통하여 데이타를 전송한다.



(그림 1) 연속미디어 저장 구조
(Fig. 1) Continuous Media Storage Structure

〈표 1〉 주변장치의 특성
〈Table 1〉 Characteristics of Peripheral Device

(디 스 크)		(CD-ROM)	
회전지연시간	11.1ms	평균탐색시간	250ms
평균탐색시간	9.4ms	캐쉬버퍼크기	64KB
캐쉬버퍼크기	512KB	데이타전송율	2.4mbps
데이타전송율	24mbps		
(SCSI) 데이타 전송율		80mbps	

(1) 디스크 수의 결정

여러 대의 디스크를 동시에 구동시키는 경우에 전체적인 데이타 검색능력은 디스크의 갯수에 비

례한다[3]. 그러나 디스크의 수를 증가시키는데는 많은 경비가 소비되므로 연속미디어 객체가 요구하는 데이타 검색 능력의 한계내에서 디스크를 증설하는 것이 바람직하다. 따라서 본 논문에서는 연속미디어 객체의 요구 대역폭과 디스크의 대역폭을 고려하여 최적의 디스크 수 M을 결정하였다.

$$M=\frac{\text{객체의 요구 대역폭 } [B_R]}{\text{디스크의 대역폭 } [B_D]}\approx 3$$

$$B_D=B_{tfr}\times\frac{size(fragment)}{size(fragment)+(T_S\times B_{tfr})}$$

여기서,

B_{tfr}=디스크의 데이타 전송율 [mbps]

size(fragment)=한번에 읽는 데이타의 크기

T_S=디스크의 (탐색시간+회전 지연시간)

(2) 디스크 시스템의 동작

각 디스크가 시스템의 요구에 의해 SCSI로부터 읽기 또는 쓰기 명령을 받았을 때 디스크의 헤드는 디스크 자체의 전송속도로 캐쉬 버퍼(cache buffer)와 데이타를 주고 받으며, 캐쉬 버퍼는 SCSI 버스의 전송속도로 SCSI와 데이타를 주고 받는다. 이때 캐쉬 버퍼에서 SCSI로, 그리고 SCSI에서 캐쉬 버퍼로 데이타의 전송이 이루어지는 시점은 캐쉬 버퍼에 데이타가 모두 찼을 때이다. 여기서 SCSI 버스의 점유 상태는 다음과 같다. 즉 읽기 동작에서 헤드의 이동이나 캐쉬 버퍼로의 데이타 전송 과정에서 SCSI 버스는 디스크로부터 분리되며, 캐쉬 버퍼의 데이타를 SCSI로 전송할 때만이 버스를 점유하게 된다. 한편 쓰기 동작에서는 쓰여질 데이타를 디스크의 캐쉬 버퍼로 전송하는 시간이 버스의 점유 시간에 해당되는데, 디스크에서의 헤드의 이동은 데이타가 전송되는 동안에도 이루어질 수 있다[5].

(그림 2)는 데이타의 읽기와 쓰기 동작에 대한 디스크 메카니즘과 SCSI 버스의 동작상태를 나타낸 것으로 각각에 소요되는 시간은 다음의 식에 의해 결정된다. 식에서 size(buffer)는 한번에 전송되는 데이타의 크기로 읽기 동작의 경우는 디스크의 캐쉬 버퍼의 크기, 쓰기 동작의 경우는 CD-ROM의 캐쉬 버퍼의 크기와 같다.

· 읽기 및 쓰기 동작의 SCSI 버스 점유시간

$$T_{RS}[sec]=\frac{size(buffer)}{B_S}$$

· 디스크에서의 읽기 및 쓰기 시간

$$T_{RW}[sec]=T_S+T_R+\frac{size(buffer)}{B_D}$$

· CD-ROM에서의 읽기 시간

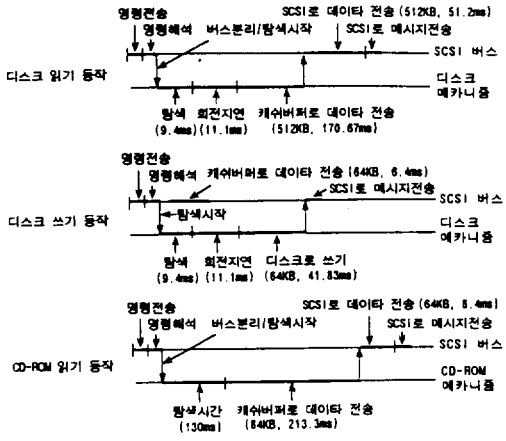
$$T_{RC}[sec]=T_S+\frac{size(buffer)}{B_C}$$

여기서,

B_S=SCSI 버스의 대역폭

T_R=디스크의 회전 지연시간

B_C=CD-ROM의 데이타 전송율



(그림 2) 디스크 메카니즘과 SCSI 버스의 동작

(Fig. 2) Operation of Disk Mechanism and SCSI Bus

그리고 (그림 3)은 하나의 CD-ROM과 3개의 디스크 사이에서의 데이타 전송 과정에 대한 알고리즘을 나타낸 것이다. 알고리즘에서 DISKn은 디스크의 번호를 뜻한다.

```
while(service is not complete) {
 parbegin
  CD-ROM_process( );
  DISK_process( );
 parend
}
CD-ROM_process( ) {
 READ data of CD-ROM disk into cache buffer;
 if(SCSI BUS!=free) Wait until BUS is free;
 else TRANSFER data of CD-ROM cache buffer to SCSI BUS;
 if(transfer!=complete) Continue transfer;
 else RELEASE SCSI BUS for other DISK or CD-ROM;
}
DISK_process( ) {
```

```
if(DISKn==ready for write) {
 TRANSFER data of SCSI BUS to DISKn cache buffer;
 if(transfer!=complete) Wait until transfer is complete;
 else RELEASE SCSI BUS and WRITE data to DISKn disk;
}
if(DISKn==ready for read) {
 READ data of DISKn disk into cache buffer;
 if(reading!=complete) Continue reading;
 else{
  if(SCSI BUS!=free) Wait until BUS is free;
  else{
   TRANSFER data of DISKn cache buffer to SCSI BUS;
   if(transfer!=complete) Continue transfer;
   else RELEASE SCSI BUS for other DISK or
      CD-ROM;
     }
   }
 }
 else check disk status;
}
```

(그림 3) 연속미디어 저장 구조의 동작
(Fig. 3) Operation of Continuous Media Storage Structure

(3) 디스크 시스템의 대역폭

연속미디어 저장구조에서 디스크 시스템의 대역폭은 3개의 디스크가 1초 동안에 인출하는 객체의 크기로 읽기 동작의 횟수와 데이타 블럭의 크기에 따라 결정된다.

이에 앞절의 식으로부터 CD-ROM에서 초당 300KB의 데이타를 연속적으로 읽어 3개의 디스크에 분할하여 저장하고, 3개의 디스크로부터 데이타를 순환적으로 읽는 경우 디스크 시스템은 62.1%의 SCSI 버스 점유율로 초당 11.6번의 읽기 동작과 4.7번의 쓰기 동작이 가능함을 알 수 있다. 그리고 이때 인출되는 객체의 크기는 47.512 mbps로, 이 값은 NTSC의 비디오 요구 대역폭인 45mbps를 초과한다.

따라서 출력장치의 데이타 처리 속도가 충분히 높고 연속미디어 객체가 디스크에 저장되어 있기만 하면 연속미디어 저장 구조는 다중의 디스크가 높은 대역폭의 버스를 공유하여 데이타를 전송함으로써 사용자의 실시간 표현에 대한 요구를 충분히 만족시킬 수 있다. 다만 객체의 각 파편들(fragments)이 디스크의 여러곳에 분산되어 있는 경우에는 헤드의 재위치(repositioning)에 따른 디스크 접근시간의 증가현상이 발생할 수 있으나, 이러한 문제는 객체의 배치기법[4], 디스크 스케쥴링[5] 등의 연구결과를 활용함으로써 해결할 수 있을 것이다.

또한 SCSI 버스의 점유율 100% 이내에서 디스크의 수를 증가시킬 경우에는 더 높은 대역폭을 갖는 연속미디어 객체의 실시간 서비스도 가능하다.

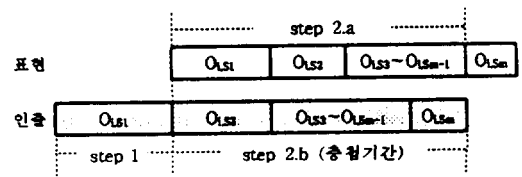
3.2 연속미디어 표현 메카니즘의 설계

(1) 객체 인출 메카니즘의 동작

연속미디어 저장구조에서 CD-ROM에 존재하는 객체에 대한 서비스 시간은 CD-ROM의 데이타를 인출하는 시간과 인출된 데이타를 출력장치를 통하여 표현하는 시간에 따라 결정된다. 그런데 만일 CD-ROM의 데이타를 인출하는 동작과 표현의 동작이 순차적으로 이루어지는 경우에 서비스 시간은 이들 두 동작의 수행에 필요한 시간의 합으로 최대가 되고, 결국 사용자는 데이타의 인출에 소비되는 시간만큼의 지연시간과 이로 인한 표현의 불연속을 감수해야 한다. 따라서 객체의 참조 요청에 대한 서비스시간을 최소화하고 연속적인 표현을 위해서는 데이타의 인출과 표현의 동작을 병렬로 수행할 수 있어야 한다.

본 논문에서 설계한 객체 인출 메카니즘은 n개의 블럭으로 구성된 하나의 객체 O_i에 대해서 인출과 표현 시간이 서로 중첩될 수 있도록 객체의 블럭들을 m개의 논리적인 블럭 $O_{i,S1}$, $O_{i,S2}$,..., $O_{i,Sm}$으로 재구성한 후, 이들 블럭들에 대해서 CD-ROM의 데이타를 디스크로 인출하는 동작과 디스크의 데이타를 표현하는 동작을 각기 다른 프로세스로 하여금 병렬로 수행하게 한다. (그림 4)는 객체 인출 메카니즘의 알고리즘과 동작 상태를 나타낸 것이다.

step 1 : get $O_{i,S1}$ from CD-ROM to $DISK_n$;
step 2 : for j=2 to m
  a) initiate representation of $O_{i,S_{j-1}}$;
  b) get $O_{i,Sj}$ from CD-ROM to $DISK_n$;
step 3 : initiate representation of $O_{i,Sm}$;



(그림 4) 객체 인출 메카니즘
(Fig. 4) Object Fetching Mechanism

(2) 객체의 재구성

객체의 재구성에 있어서 마지막 논리 블럭 $O_{i,Sm}$의 크기는 표현과 인출의 중첩기간을 최대로 하기 위해 연속미디어 객체의 구성 단위인 단일 블럭으로 결정한다. 그리고 첫번째 논리 블럭 $O_{i,S1}$의 크기는 객체 O_i의 전체 크기에서 step2,a의 지속 기간(duration)동안 디스크로 인출된 객체의 모든 크기를 제외한 나머지로 결정한다. 마지막으로 두번째 이후의 블럭들 $O_{i,S2},\ldots,O_{i,Sm-1}$의 크기는 step2.b의 지속 기간동안에 디스크로 인출된 객체 전체의 크기를 표현에 요구되는 대역폭과 CD-ROM의 대역폭에 따라 분할한다.

$$size(O_{i,Sm}) = size(block) \quad (1)$$

$$size(O_{i,S1}) = size(O_i) - \sum_{j=2}^{m} size(O_{i,Sj})$$

$$= \{n - \frac{B_C}{B_R} \times (n-1)\} \times size(block) \quad (2)$$

$$size(O_{i,Sj}) = \frac{B_C}{B_R} \times size(O_{i,Sj-1})$$

$$(2 \le j \le m-1) \quad (3)$$

이와같이 객체를 재구성하여 인출과 표현의 동작을 실시하는 경우 사용자의 서비스 요구에 대한 시작 지연시간은 (그림 4)의 step1의 지속기간에 해당되는데, 이는 CD-ROM의 객체 $O_{i,S1}$에 해당하는 위치로 읽기 헤드를 위치하는 시점에서부터 $O_{i,S1}$을 디스크로 적재 완료하는데 까지 걸리는 시간이다. 각 객체의 시작 지연시간 $T_L(O_i)$를 수식으로 표현하면 다음과 같다.

$$T_L(O_i) = \frac{size(O_{i,S1})}{B_C}$$

$$= \frac{1}{B_C}\{n - \frac{B_C}{B_R} \times (n-1)\} \times size(block) \quad (4)$$

(3) 디스크 공간 확보 알고리즘

CD-ROM의 객체를 인출하여 디스크로 적재하는 과정에서 만일 디스크의 공간이 부족한 경우, 시스템은 인출되는 객체를 위한 충분한 공간을 확보하기 위해 디스크에 존재하는 객체의 일부 또는 전부를 삭제하여야 한다. 이때 객체의 삭제는 재참조 요청에 대한 실기율(miss ratio)과 시작 지연시간의 변화를 최소화하는 방향으로 실시되어야 한다.

본 논문에서 제안한 디스크 공간 확보 알고리즘은 인출되는 객체를 위한 충분한 공간이 확보될 때까지 3단계의 절차를 순차적으로 수행한다. 먼저 제1단계에서는 디스크에 존재하는 각 객체 O_i의 파편들에서 첫번째 논리 블럭을 제외한 나머지 부분을 삭제한다. 이것은 (식 4)에서와 같이 각 객체의 참조 요청에 대한 시작 지연시간을 최소화한다. 만일 제1단계에서 충분한 디스크 공간을 확보하지 못한 경우에는 제2단계로 넘어가게 되는데, 제2단계에서는 디스크에 존재하는 각 객체의 파편들의 최소 부분을 제외한 나머지 부분을 삭제한다. 이 단계의 결과는 가능한 많은 수의 객체를 디스크에 존재케 함으로써 이들 객체를 재참조하는 각 요청들에 대한 적중률(hit ratio)을 높여 지연시간의 변화를 감소시킨다. 마지막 제3단계는 최악의 경우에 있어서 디스크 공간의 확보를 목적으로 하는 것으로서 디스크에 존재하는 객체들중 참조 확률이 가장 낮은 것의 일부 또는 전부를 삭제한다.

디스크 공간 확보 알고리즘의 각 단계의 동작에 있어서 공통적인 사항은 디스크에 존재하는

```
{
  for(P(O_i)=lowest; P(O_i)=highest; select P(O_i))   //step-1
  {
    if(DISK(O_i)>size(O_i,S_1)){
      delete_space=DISK(O_i)-size(O_i,S_1);
      DISK(O_i)=size(O_i,S_1);
      free_space=free_space+delete_space;
    }
    if(free_space==required_space) break;
  }
  for(P(O_i)=lowest; P(O_i)=highest; select P(O_i))   //step-2
  {
    if(DISK(O_i)>LEAST(O_i)){
      delete_space=DISK(O_i)-LEAST(O_i);
      DISK(O_i)=LEAST(O_i);
      free_space=free_space+delete_space;
    }
    if(free_space==required_space) break;
  }
  for(P(O_i)=lowest; P(O_i)=highest; select P(O_i))   //step-3
  {
    delete_space=DISK(O_i);
    DISK(O_i)=0;
    free_space=free_space+delete_space;
    if(free_space==required_space) break;
  }
}
```

(그림 5) 디스크 공간 확보 알고리즘
(Fig. 5) Algorithm for Securing Disk Space

객체의 논리 블럭들에 대해서 $O_{i,Sm}, O_{i,Sm-1}, \ldots, O_{i,S1}$의 순서로 삭제를 실시하며, 삭제된 공간에 새로이 요청되는 객체를 $O_{i,S1}, O_{i,S2}, \ldots, O_{i,Sm}$의 순서대로 적재한다. 그리고 각 객체의 참조 확률이 낮은 것에서 높은 것의 순서대로 삭제를 실시한다. (그림 5)는 디스크 공간 확보 알고리즘을 나타낸 것이다.

(4) 삭제 대상 객체의 크기 결정

디스크 공간 확보 알고리즘의 동작에 있어서 제1단계의 첫번째 논리 블럭의 크기는 (식2)에 따르며, 제2단계의 각 객체의 최소 부분은 다음과 같이 결정한다.

즉, $1 \le i \le m$에 대해서 한 주기의 시간동안 확률 $P(O_i)$로 참조 요청이 발생하는 m개의 객체 $O_1, O_2, \ldots, O_m$이 있을 때, 이들 객체에 대한 지연시간의 변화(variance)는 각 객체의 시작 지연시간과 참조 요청된 확률에 따라 달라지므로[11] 최소부분을 $LEAST(O_i)$라 하고 (식5)와 같이 정의한다.

$$LEAST(O_i) = k \times P(O_i) \times size(O_{i,S1}) \qquad (5)$$

여기서,

$$\sum_{i=1}^{m} P(O_i) = 1, \qquad (0 \le P(O_i) \le 1)$$

$$\sum_{i=1}^{m} LEAST(O_i) \le size(disk)$$

$size(disk)$ = 연속미디어에 할당된 디스크 공간

그리고 한 주기 동안 참조 요청된 m개의 객체의 첫번째 블럭의 평균 크기와 평균 참조 확률, 그리고 각 객체의 최소부분의 평균을 구하면 다음과 같다.

$$size(O_{i,S1})' = \sum_{i=1}^{m} \{P(O_i) \times size(O_{i,S1})\}$$

$$P(O_i)' = \frac{\sum_{i=1}^{m} P(O_i)}{m} = \frac{1}{m}$$

$$LEAST(O_i)' = k \times P(O_i)' \times size(O_{i,S1})'$$

이때 이상적인 경우는 m개의 객체의 최소부분이 모두 디스크에 존재할 때이므로, 위의 식들로부터 다음의 관계를 유도할 수 있다.

$$size(disk) = m \times LEAST(O_i)'$$

$$= m \times k \times \frac{1}{m} \times size(O_{i,S1})'$$

$$k = \frac{size(disk)}{size(O_{i,S1})'} \qquad (6)$$

따라서 (식6)을 (식5)에 대입함으로써 지연시간의 변화를 최소화하는 $LEAST(O_i)$가 최종적으로 결정되고, 이로부터 제2단계의 결과에 의한 시작 지연시간을 구할 수 있다.

$$T_L(O_i) = \frac{1}{B_C} \times \{size(O_{i,S1}) - LEAST(O_i)\}$$

4. 성능평가 및 분석

연속미디어 저장 구조에 대해서는 제3장의 해석을 통하여 디스크 시스템의 대역폭이 NTSC의 비디오 요구 대역폭을 초과하는 것으로 확인되었다. 따라서 이 절에서는 연속미디어 표현 메카니즘의 적용이 표현의 시작 지연시간과 시작 지연시간의 변화 감소에 미치는 영향을 시뮬레이션 결과를 통하여 확인한다. 이때의 시뮬레이션 모델은 다음과 같이 설정하였다.

1) 한 번에 서비스할 수 있는 미디어의 수는 하나이다.
2) 각 실험에서 객체의 요구 대역폭은 45mbps, 22.5mbps, 11.25mbps, 5.625mbps의 네가지 경우를 선택적으로 적용한다.
3) CD-ROM의 대역폭은 2.4mbps이며, 데이타베이스에 100개의 연속미디어 객체가 존재한다.
4) 시작 지연시간의 변화 평가에서 각 객체의 크기 및 참조 확률은 평균(mean)이 15인 지수분포에 따르는 것으로 가정한다.
5) 연속미디어를 위한 디스크 공간의 크기는 참조 요청된 모든 객체들의 첫번째 블럭의 합에 해당하는 값과, 이 값의 1/2, 1/4 값을 선택적으로 적용한다.

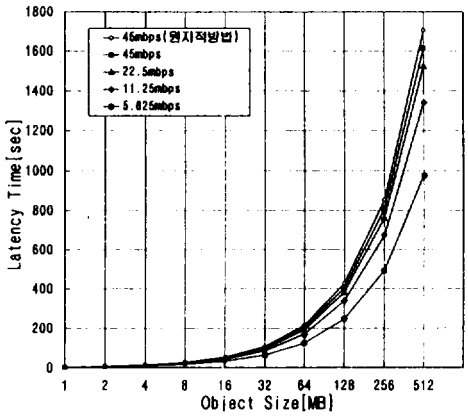
4.1 시작 지연시간의 평가

객체 인출 메카니즘의 실효성을 확인하기 위하여 객체의 인출과 표현이 순차적으로 이루어지는 원자적인(atomic) 방법과 객체 인출 메카니즘을 적용했을 때의 시작 지연시간의 차이를 구하였

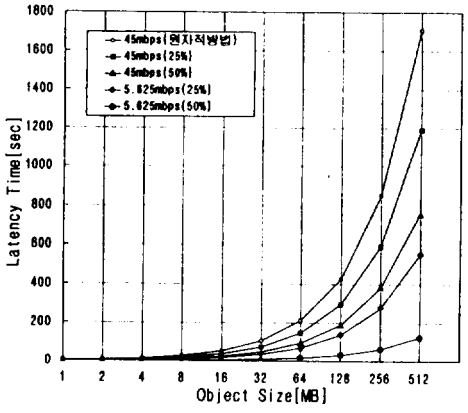
다. 이 실험에서는 시작 지연시간의 비교를 목적으로 하므로 각 객체의 참조확률과 디스크 공간의 크기는 고려하지 않았다.

(그림 6)은 첫번째 실험으로 단일 객체의 크기를 1MB에서 512MB까지 변화시켰을 때의 시작 지연시간을 나타낸 것이다. 그림에서와 같이 객체의 요구 대역폭이 낮을수록, 그리고 객체의 크기가 클수록 원자적인 방법에 비해 시작 지연시간이 감소함을 알 수 있는데, 객체의 크기가 512MB이고 요구 대역폭이 45mbps와 5.625mbps일 때 각각 5.3%, 42.6%의 시작 지연시간 감소가 발생하였다.

시작 지연시간 평가의 두번째 실험에서는 참조 요청되는 객체의 일부가 디스크에 적재되어 있는



(그림 6) 객체 크기에 따른 지연시간(1)
(Fig. 6) Latency Time versus Object Size(1)



(그림 7) 객체 크기에 따른 지연시간(2)
(Fig. 7) Latency Time versus Object Size(2)

경우의 시작 지연시간을 원자적인 방법과 비교하였다. (그림 7)은 그 결과를 나타낸 것으로 객체의 크기가 512MB일 때, 요구 대역폭이 45mbps이고 객체의 25%와 50%가 이미 디스크에 적재되어 있는 경우 원자적인 방법에 비하여 각각 30.3%와 55.3%의 시작 지연시간 감소가 발생하였다. 또한 요구 대역폭 5.625mbps에 대해서는 67.6%와 92.6%까지 시작 지연시간이 감소되었는데, 이들 값은 (그림 6)의 실험 결과에 비하면 상당한 수준의 시작 지연시간 감소에 해당된다.

결국 이 두가지 실험에서 객체 인출 메카니즘의 적용은 시작 지연시간의 변화를 감소시키며, 객체의 일부가 디스크에 존재하는 경우에는 감소의 효과는 더욱 커짐을 확인할 수 있다. 따라서 높은 대역폭을 갖는 연속미디어 객체들을 계속하여 서비스하는 경우에 객체 인출 메카니즘은 초기의 한번의 디스크 적재 과정을 수행함으로써 재참조 요청되는 객체들의 표현의 시작 지연시간과 이로 인한 불연속을 감소시킬 수 있다.

4.2 지연시간의 변화 평가

앞의 실험에서 각 객체의 일부가 디스크에 적재되어 있을 때 연속 미디어 표현의 시작 지연시간이 감소됨을 확인하였다. 따라서 이 절에서는 제한된 디스크 공간에 객체를 최적으로 적재하는 방법을 실험을 통하여 구하였다.

이 실험에서는 디스크 공간 확보 알고리즘을 실행하였을 때의 각 객체에 대한 시작 지연시간

〈표 2〉 지연시간의 변화
〈Table 2〉 Variance of Latency Time

디스크 공간	B_R=45mbps, C=291MB		B_R=5.625mbps, C=191MB	
	k	σ	k	σ
C/4	2.93	11.84	2.94	7.70
	5.85	9.07	5.88	5.86
	8.78	3.62	8.82	2.26
	11.70	5.90	11.76	3.73
	14.63	9.00	14.70	5.91
C/2	5.85	10.45	5.88	6.87
	11.70	9.39	11.76	6.15
	17.56	6.60	17.64	4.32
	23.41	8.56	23.52	5.68
	29.26	10.26	29.40	6.78
C	11.70	0	11.76	0
	23.41	0	23.52	0
	35.11	0	35.28	0
	46.82	0	47.04	0
	58.52	0	58.80	0

들의 표준편차를 계산하고, 이 알고리즘의 제2단계에서 결정한 각 객체의 최소 부분이 시작 지연시간의 변화를 최소화함을 확인하였다. 이때 입력 파라메타는 시뮬레이션의 조건에서 설명한 것과 같으며, k의 값은 편의상 (식 6)에 의한 값을 기준으로 ±33.3%, ±66.6%씩 변화시켰다.

〈표 2〉는 계산된 결과를 나타낸 것으로, C는 참조 요청된 모든 객체들의 첫번째 블럭의 합을 나타낸 것이다. 계산결과 객체의 요구 대역폭이 45mbps이고, 디스크 공간의 크기가 C/4인 경우에 (식 6)에 의한 값인 k=8.78은 k=2.93에 비하여 69.4%의 표준편차 감소가 발생하였다. 그리고 k=14.63에 비해서는 60%의 표준편차 감소를 나타내었다. 또한 동일한 대역폭에 대해서 디스크 공간의 크기가 C/2인 경우에 k=17.56은 k=5.85와 k=29.26에 비해서 각각 36.9%와 35.7%의 표준편차 감소를 보였다. 이 결과는 디스크 공간이 C/4일 때 보다 낮은 감소 비율에 해당되는데, 이것은 디스크의 공간이 작을수록 시작 지연시간의 변화에 대한 감소효과가 커짐을 증명하는 것이다.

한편 디스크 공간의 크기가 C인 경우에는 k에 관계없이 표준 편차가 0의 값을 가지게 되는데, 이는 각 객체의 첫번째 블럭이 디스크에 존재하는 경우에는 시작 지연시간이 발생하지 않음을 나타내는 것이다.

결과적으로 디스크 공간 확보 알고리즘은 제한된 디스크 공간에 가능한 많은 수의 객체를 적재하면서 시작 지연시간의 변화를 최소화하는 수단을 제공한다.

5. 결 론

본 논문에서는 저장장치의 대역폭을 높이기 위한 방법으로 SCSI와 다중의 디스크 그리고 CD-ROM으로 구성된 저장 구조를 제안하였다. 그리고 제안된 저장 구조에서 CD-ROM의 연속미디어를 인출하여 표현하는 동작을 병렬로 수행함으로써 표현에 대한 시작 지연시간과 불연속을 최소화하는 메카니즘을 설계하였다. 또한 객체의 인출 동작시 가능한 많은 수의 객체를 디스크에 존재케 함으로써 각 객체의 재참조 요청에 대한 실기율과 지연시간의 변화를 최소화하는 디스크 공간 확보 알고리즘을 설계하였다.

제안한 저장 구조를 해석한 결과 디스크 시스템의 대역폭은 47mbps로 NTSC의 비디오 요구 대역폭을 초과하였으며, 객체 인출 메카니즘의 적용은 객체의 인출과 표현이 순차적으로 이루어지는 방법에 비하여 시작 지연시간이 감소함을 확인하였다. 이때 참조 요청된 객체의 일부가 디스크에 적재되어 있는 경우에는 적재된 크기에 비례하여 감소효과가 증가함을 알 수 있었다. 또한 각 객체의 참조 확률과 평균 크기를 고려하여 적재 대상 객체의 크기를 결정한 값이 지연시간의 변화를 최소화함을 확인하였다.

따라서 본 논문의 결과는 연속 미디어 객체에 대해서 참조 요청이 자주 발생하는 경우 서비스의 응답시간을 줄이고, 표현의 불연속을 방지할 수 있는 방법을 제공하므로 연속미디어의 실시간 검색 또는 이들 미디어의 동기화를 위한 연구에 활용이 가능할 것으로 기대한다. 향후 본 연구는 디스크상에서의 객체의 배치, 서비스 스케쥴링 등에 관련된 연구와 접목이 이루어질 예정이다.

참 고 문 헌

[1] D.James Gemmell and Jiawei Han, "Delay-Sensitive Multimedia on Disks", IEEE Multimedia, pp.56-67, Fall. 1994.

[2] P.V. Rangan, H.M. Vin, and S. Ramanathan, "Designing an on-demand multi-media service", IEEE Computer, pp.56-65, Jul 1992.

[3] Shahram Ghandeharizadeh, "Continuous Retrieval of Multimedia Data Using Pa-rallelism", IEEE Transactions on Knowledge & Data Engineering, VOL. 5, No. 4, pp.658-669, August 1993.

[4] Steven Berson, Shahram Ghandeharizadeh, Richard Muntz, Xiangyu Ju, "Staggered Striping in Multimedia Information Systems", In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.79-90, 1994.

[5] Chris Ruemmler and John Wilkes, "An Introduction to Disk Drive Modeling", IEEE Computer, pp.17-28, march 1994.

[6] 황기태, "고성능 디스크 입출력을 위한 디스크 스케쥴링 알고리즘", 서울대교 대학원 공학박사학위논문, 1994.

[7] P. Venkat Rangan and Harrick M. Vin, "Efficient Storage Techniques for Digital Continuous Multimedia", IEEE Transactions on Knowledge and Data Engineering, VOL.5, No. 4, pp.564-573, August 1993.

[8] 서근주, 백윤철, 고건, "연속 미디어 검색을 위한 최저 작업량 할당 기법", '94 한국정보과학회 가을 학술발표논문집, 제21권, 제2호.

[9] Richard Staehli and Jonathan Walpole, "Contained-Latency Storage Access", IEEE Computer, March 1993

[10] Raymond T. Ng and Jinhai Yang, "Maximizing Buffer and Disk Utilizations for News On-Demand", In Proceedings of the International Conference on Very large databases, pp. 451-462, 1994.

[11] 이동한, 이성덕, 황강진, '시뮬레이션 이론과 실제', 교학사, pp. 23-55, 1993.

여 인 국

1984년 명지대학교 전자공학과 졸업(학사)
1995년 경원대학교 대학원 전자계산학과(석사)
1984년~91년 국방과학연구소 연구원
1991년~현재 생산기술연구원 선임연구원
관심분야 : 멀티미디어시스템, 영상처리, 컴퓨터조직 응용

황 대 훈

1977년 동국대학교 수학과 졸업(학사)
1983년 중앙대학교 대학원 전자계산학과(석사)
1991년 중앙대학교 대학원 전자계산학과(박사)
1983년~85년 한국산업경제기술연구원(KIET) 연구원
1987년~현재 경원대학교 전자계산학과 부교수
1995년~현재 경원대학교 전자계산소장
관심분야 : 멀티미디어 시스템, CAD/CAM, 분산 시스템 소프트웨어 등