

VOD 시스템에서의 디스크 스케줄링 기법의 성능 평가

전 용 희[†] · 이 상 학^{††}

요 약

일반적으로 주문형 비디오 시스템(Video-On-Demand)에서는 계산 능력보다는 I/O 기능 및 저장 요구사항이 더 중요하다. CPU 기술의 발달로 프로세서의 성능과 디스크 속도 사이의 차이는 점차로 넓어지고 있는 실정이다. 그러므로, 디스크 시스템이 VOD 시스템에서 대표적으로 병목 현상을 초래하게 된다. 이러한 본리적인 제한을 완화하기 위하여 디스크 배열 시스템이 사용된다. 디스크 배열 시스템의 I/O 성능은 프로세서의 능력, I/O 스케줄링 정책, 배열에 사용되는 구성 디스크의 수, 디스크 블록 크기, 블록 레일 정책 등에 의하여 개선될 수 있다.

본 논문에서 고려된 디스크 스케줄링 기법들로는 EDF(Earliest Deadline First), SCAN, SCAN-EDF, Round-robin, GSS(Grouped Sweeping Scheme) 및 SCAN의 변형된 형태인 C-SCAN, LOOK, C-LOOK 등이 있다. 이러한 디스크 스케줄링 기법들을 중심으로 디스크 배열 시스템의 성능을 비교·검토하기 위하여 여러 가지 관련 파라미터를 고려하여 다양한 환경 하에서 모의실험을 수행하고 파라미터들이 성능에 미치는 영향을 분석하였다.

Performance Evaluation of Disk Scheduling Schemes in a VOD System

Yong-Hee Jeon[†] · Sang-Hag Lee^{††}

ABSTRACT

In a VOD(Video-On-Demand) system, I/O performance and storage requirements are generally more important than computing capability. Due to the development of CPU technology, the gap between the performance of processors and disk speed are widening. Therefore, the disk system typically becomes the bottleneck in a VOD system. In order to alleviate those effects from physical constraints, disk array system is used. The I/O performance of disk arrays can be improved by the capability of processors, I/O scheduling policy, the number of member disks in the array, disk block size, block placement method etc.

In this paper, we considered the disk scheduling schemes including EDF(Earliest Deadline First), SCAN, SCAN-EDF, Round-robin, GSS(Grouped Sweeping Scheme), and C-SCAN, LOOK, C-LOOK which are variations of SCAN. Mainly focusing on those disk scheduling schemes, we performed the simulation in order to compare and review the performance of disk arrays considering several related parameters under diverse environments and analyzed the effects of those parameters to the performance.

1. 서 론

주문형 비디오 시스템의 스트림 데이터를 구성하는

오디오 샘플 및 비디오 프레임은 연속적으로 전달되는 연속 매체(continuous media)이다. 연속 매체는 정해진 시간 내에 연속하여 제시될 때에만 의미를 가진다. VOD 시스템에서 연속적인 데이터 흐름을 보장하기 위하여 매체를 연속적으로 검색을 하여야 하며 서지는 출력 장치에서 기아(starvation)를 방지할 수 있도록

[†] 강희원, 대구가톨릭대학교 컴퓨터정보통신공학과 교수
^{††} 장희원, 영남대학교 의료원 생체의공학과
논문접수: 2000년 2월 11일, 심사완료: 2000년 10월 17일

디스크(즉, 저장소)로부터 충분한 데이터 스트림 버퍼에 공급하여야 한다.

VOD 서버는 비디오 저장소 및 클라이언트의 요청에 따라 저장소에 있는 비디오 데이터를 읽어서 네트워크로 보내주는 비디오 펌프를 가질 수 있다. 마이크로 프로세서로 구성된 비디오 펌프의 속도에 비해 저장소에서 해당되는 데이터를 검색하는 속도가 훨씬 느리므로 비디오 펌프의 이용률은 저장소의 검색성능에 의하여 제한되고 최대 수용 스트림 수 또한 저장소의 검색 성능에 의해 결정된다. 비디오 저장소의 성능이 물리적인 특성에 의해 제한되므로 결국 비디오 저장소에서의 검색 성능이 VOD 시스템 전체 성능에 한계로 작용할 수 있다.

동시수용 스트림 수를 최대화하기 위한 방법은 디스크 시스템의 물리적인 성질, 디스크 배열의 구성 방법 및 디스크 스케줄링(disk scheduling) 기법과 관련이 있다. VOD 시스템에서 스케줄링 기법은 동시 수용 스트림들이 모두 동일한 데드라인(deadline)을 가지도록 설계하는 Round-based scheduling)과 동일한 데드라인 없이 스트림 각각의 읽기 데드라인을 가지도록 설계하는 데드라인 기반 스케줄링(Deadline-based scheduling)으로 분류할 수 있다. 즉, 비디오 검색에서 액세스 정책은 라운드별로 읽어야 할 데이터 크기를 정하며 스케줄링 기법은 스트림들의 서비스 순서를 결정한다. VOD 시스템에서는 이들 두 가지 측면을 동시에 고려하여야 전체 버퍼 크기, 시작 지연, 지연 지터와 같은 요소들이 결정될 수 있다.

이런 메체 스트림의 연속 검색은 데드라인(deadline)을 가진 일련의 주기적인 태스크로 이루어져 있다. 여기서 태스크는 디스크로부터의 비디오 검색에 해당된다. 검색은 비스터하기 때문에 미디어 블록들은 전송 전에 버퍼될 필요가 있다. 단일 스트림을 효율적으로 서비스하는 문제는 다음과 같은 세 가지 면에서의 문제가 된다[1].

- 1) 시청자 버퍼의 기이로부터의 방지
- 2) 버퍼 공간의 최소화
- 3) 시작(start) 지연의 최소화

다수의 스트림이 인기 있는 영화와 같은 동일한 과일을 접근할 때에도 다른 스트림들이 동시에 그 과일의 다른 부분에 접근하게 된다. 디스크 데이터율이 단일 스트림 데이터율을 훨씬 초과하기 때문에(예를

들어, 3~4MB/s는 0.42MB MPEG-2 데이터율을 초과한다), 동시에 서비스될 수 있는 스트림 수는 다수의 스트림들을 디스크 헤드에서 다중화 함으로써 일반적으로 증가한다. 그러므로, 주의 깊은 디스크 요구 스케줄링이 필요하다.

본 논문에서는, 디스크 배열 구조에 대해 디스크 스케줄링 기법과 여러 가지 파라미터들이 디스크 성능에 미치는 영향을 분석하고자 모의실험을 수행하고 그 결과를 제시하였다. 그리고 데이터 액세스 정책인 CTL(Constant Time Length)과 CDL(Constant Data Length)을 이용하여 수용할 수 있는 스트림 수를 시뮬레이션에 의하여 구하고 두 개의 혼합 스트림에 대한 스케줄러블(schedulable) 영역을 구하였다.

2. 주문형 비디오(VOD) 시스템

VOD 시스템은 클라이언트/서버 구조로서 클라이언트는 비디오의 검색을 서버에 요청한다. 서버는 클라이언트의 요청이 수신되면 해당되는 데이터를 찾아 클라이언트에 서비스를 수행해주게 된다. 이때는 방대한 데이터량 때문에 시간별로 일정량의 데이터를 읽어 내주는 라운드 기반의 서비스를 기본 방식으로 사용한다. 이러한 시스템에서는 동시에 여러 스트림을 서비스해야 하더라도 각각의 서비스 품질(QoS: Quality of Service)을 만족시켜야 하므로 자원의 할당 및 관리가 중요한 이슈이다.

VOD 시스템에 대한 모델링 시 고려되는 성능지수는 네트워크 지원인 대역폭 할당과 버퍼 구성비용, 지연 및 지연변이와 같은 것들이 있다. 또 하나 중요한 지수로는 동시 수용 스트림 수의 향상 정도이다.

버퍼를 중심으로 봤을 때 라운드 서비스를 제공하는 VOD 시스템의 시간에 따른 변화는 생산자-소비자 모델(Producer-Consumer Model)을 사용하여 나타낼 수 있다. 생산에 해당하는 버퍼에 도착하는 데이터량을 $A(t)$ 로, 소비에 해당하는 버퍼에서 재산을 위해 읽어 가는 데이터량을 $C(t)$ 로, 비퍼링되어야 하는 양을 $B(t)$ 로 나타내면, $B(t)$ 는 $A(t)$ 와 $C(t)$ 의 차이가 된다[2].

도착량은 소비량보다 항상 크거나 같아야 한다. 그렇지 않으면 버퍼기근 현상이 생겨 화질저하 현상이 발생한다. 하지만 도착률이 소비율에 비해 너무 크다면 버퍼 차지량(buffer occupancy)이 증가하여 버퍼 오버플로 현상이 발생한다. 따라서 전송률이 소비율과

적절히 맞을 수 있도록 전송 측에서는 울 제어가 반드시 행해져야 하고 통신 시스템의 다른 모든 부분들도 이 울로 데이터를 전달해야 한다

만만 도착률이 소비 울보다 낮다면 버퍼기근 현상이 발생한다. 이 경우는 초기지연이 매우 길어지게 된다 또한 시스템 응답시간이 길게 되어 초기버퍼가 매우 커지게 된다. 그러므로 평균 도착률은 스트림이 연속적으로 재생될 수 있도록 소비 울과 거의 보조를 맞추어야 한다 또한 전송 측은 소비 울로 데이터를 보내야 하고 종단간 전송 대역폭이 보장되어야 한다.

또 하나의 중요한 자원으로 네트워크 대역폭이 있다 대역폭은 시간당 채널 용량으로 생각할 수 있다. 대역폭은 보다 많은 스트림들이 유한한 자원을 동시에 액세스한다는 사실 때문에 중요한 자원이 된다. 미시적 관점에서는 이러한 대역폭은 충분한 것으로 가정할 수 있지만, 거시적인 관점에서는 end-to-end 성능분석이 대상이므로 대역폭이 주요한 자원일 수 있다. 특히 MPEG-2로 압축된 비디오 데이터는 버스트성이 큰 트래픽으로 QoS를 보장하기 위해 필요 이상으로 많은 대역폭을 할당할 경우는 대역폭 이용률에서뿐만 아니라 수용 스트림수 측면에서도 많은 성능저하 요인이 될 것이다.

MPEG-2로 압축된 데이터는 원래 VBR적인 특성을 가진다. 그러나 이들을 네트워크를 통해 보낼 때는 VBR특성을 고려하여 보내거나 아니면 CBR로 만들어 보낼 수 있다. 그러므로 대역폭을 할당하는 방법은 CBR적인 방법과 VBR적인 방법 또는 이 둘을 결합한 semi-static 방법 등이 존재한다.

지연 및 지연변이 요구사항 또한 비디오와 같은 실시간성을 요구하는 트래픽을 다룰 때에는 중요한 성능 지수이다. 버퍼크기는 도착 데이터울에 의해 다중화되는 버퍼링 시간과 같다. 초기 버퍼링 시간은 최대 패킷 지연 지터와 같으므로 지연지터가 크면 클수록 버

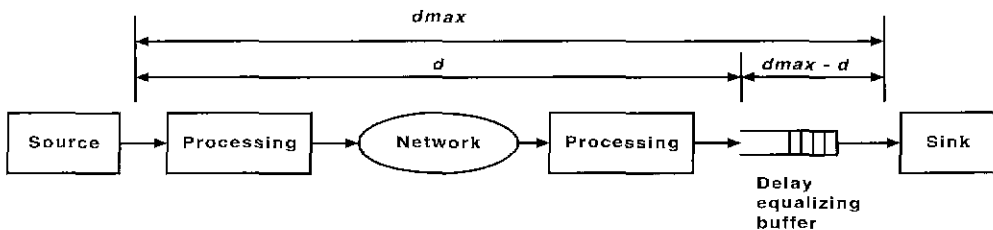
퍼 요구량은 많아지게 된다[3]

지연지터는 패킷처리시간의 차이, 네트워크 액세스 시간의 차이, 큐잉지연 차이와 같은 요소에 의해 발생한다 지연지터는 목적지에 있는 FIFO 버퍼에 의해 재생 전에 제거된다. 이러한 버퍼링 기법의 원리는 가변적인 양의 지연을 각 패킷에 부가하여 소스로부터 싱크로의 각 패킷에 대한 전체 지연이 같도록 하는 것이다.

(그림 1)에서 지연이 d 인 패킷은 $d_{max}-d$ 동안 버퍼링되었으므로 d_{max} 와 같은 전체 지연을 가지게 된다 [2]. 멀티미디어 통신 요구를 충족시키기 위해서 버퍼는 오버플로 되어도 안되고 기근현상이 발생해서도 안 된다. 그리고 버퍼크기는 너무 크지 않아야 한다 버퍼 크기가 크다는 것은 종단간 지연이 크다는 것을 의미하며 시스템의 비용이 많이 든다는 것을 의미한다.

최대수용 스트림수 향상을 위해 멀티캐스트 기능을 이용하기 위한 배칭기법의 사용 및 서버 저장소의 핫스팟 해소를 위한 내용의 효율적인 구성기법을 고려할 수 있다 요즈음엔 인터넷 멀티캐스팅 기능을 지원하도록 IP 어드레스를 새롭게 구성하려고 노력하고 있으며 ATM 기술은 자기 경로배정(self-routing) 기능을 가지고 있어서 이러한 멀티캐스팅 기능의 구현에 대한 활발한 연구가 이루어지고 있다 배칭기법은 인기 있는 내용의 요청을 여러 개 한 단위로 묶어서 네트워크로 내보내어 저장소로의 요청을 줄여서 대역폭을 절약 하자는 의도로 만들어진 것이다. 물론 이때는 네트워크가 멀티캐스팅 기능을 지원한다는 선제조건 하에이다 부하를 줄이기 위한 이러한 기법은 저장소의 구성 측면에서도 고려되고 있다. 다시 말하자면 보다 많은 요청을 수용하기 위해 같은 복사내용을 여러 분산 노드에 저장시키던가 하나의 내용을 여러 노드에 나누어 분산해서 저장하는 방법을 고려할 수 있다[4].

마지막으로 언급해야 할 문제는 QoS 문제이다. 이것은 VOD 서버의 특성상 동시수용 스트림수의 향상



(그림 1) VOD 시스템에서 지연 원인의 요소

을 보장하면서도 기존에 서비스되던 스트림의 QoS를 보장해야 하기 때문이다. 그러므로 해당 QoS에 맞는 승인제어기법이 필요하며 수용된 서비스 내용이 기존에 서비스받는 내용의 QoS에 영향을 미쳐서는 안 된다[5].

3. 디스크 배열 성능

디스크 배열은 단일 디스크들의 물리적인 한계를 병렬성을 이용하여 극복하고 멀티미디어의 대용량성을 만족시키기 위해 제안된 것으로 물리적으로는 헤드와 원하는 트랙으로 움직여 원하는 데이터를 단일 채널을 통해서 전송해내는 기존의 구조를 따르지만, 논리적으로는 이들 단일 디스크들을 효율적으로 묶는 방법을 모색하여 디스크의 성능을 최대화하는 기술을 필요로 한다. 디스크 배열의 구조를 분석하기 위하여 단일 디스크의 특성을 검토하기 위한 측면과 이들을 디스크 배열로 효율적으로 구성하기 위한 측면으로 나누어 기술할 필요가 있다. 이들을 디스크 시스템의 물리적인 관점과 논리적인 관점이라 한다[6].

디스크 배열의 물리적인 관점과 논리적인 관점에서 본다면 물리적인 관점에서의 성능 결정 요소들은 단일 디스크의 액세스와 관련되는 탐색시간, 회전지연시간, 전송시간 및 부가적인 오버헤드들이 포함되며, 논리적인 관점에서의 성능 결정요소들은 디스크 배열을 구성하기 위해 필요한 여러 가지 파라미터 즉, 디스크 배열의 멤버 디스크 수, 동기화 정도(ds : degree of synchronization), 데이터 입상성(granularity) 등이 성능 결정 요소가 된다.

3.1 물리적인 관점

물리적인 관점에서의 성능 결정 요소들은 주로 단일 디스크에서의 데이터 액세스 과정과 관련된다.

디스크의 데이터 검색에 소요되는 시간(T_{search})은 식 (1)과 같은 요소로 구성된다.

$$T_{search} = T_{seek} + T_{rot-lat} + T_{transfer} + T_{hit-stitch} \quad (1)$$

여기서, T_{seek} 는 탐색하고자 하는 실린더로 디스크 암을 움직이는 탐색시간을 말하며, $T_{rot-lat}$ 은 탐색과정이 끝나고 원하는 데이터가 헤드 밑을 지나갈 때까지 기다리는 회전지연시간 그리고 $T_{transfer}$ 은 디스크 헤드

가 디스크 플래터의 표면으로부터 데이터를 읽고 이 데이터를 시스템 내부나 디스크 제어 장치의 디스크 캐쉬 영역으로 전송하는 데이터 전송시간, 또 $T_{hit-stitch}$ 는 헤드 전환시간을 의미한다.

탐색시간(seek time)은 디스크 성능에 영향을 미치는 가장 중요한 요소이다. 탐색 시간은 회일 검색의 접근 형태, 동시에 접근하는 다른 화일의 수 및 검색되는 디스크 블록의 입상성(granularity)에 영향을 받는다. 탐색시간 모델은 트랙의 선형함수(linear function of tracks)로 나타내는 방법과 트랙의 비선형 함수(nonlinear function of tracks)로 나타내는 방법이 있다[7].

디스크의 1회전 시간이 L 일 때 회전지연 시간은 $[0, L]$ 사이의 균일 분포를 가지며 평균 $L/2$ 시간을 소요한다. 즉, 디스크 회전 속도가 3600rpm이면 한번 회전에 16.7ms가 걸리며 평균 회전지연 시간은 8.35ms이다. 그러므로 회전지연(rotational latency) 시간은 데이터 전송시간과 함께 디스크 회전속도와 관계가 있다. 현재 대표적인 속도는 3600rpm이지만 7200rpm까지 서비스가 가능하다.

하나의 디스크가 요구 크기 s 블록을 전송하기 위해서 걸리는 전송 시간(transfer time)은 식 (2)와 같다.

$$T_{transfer} = \frac{s}{D_{disk}} \quad (2)$$

여기서, D_{disk} 는 디스크 전달률이다. 이러한 전송시간은 디스크 회전속도 및 디스크의 저장밀도 모델링 방법에 직접적인 영향을 받으며 이러한 두 가지 요소는 디스크의 물리적인 제한요소로 작용한다.

3.2 논리적인 관점

논리적인 관점은 주로 디스크 배열 구성과 관련된이다. 그러한 요소들에는 디스크 배열을 구성하는 멤버 디스크의 수, 디스크 동기화 정도, 스트라이핑(striping)된 데이터 블록의 크기와 관련되며, 이 때 블록은 읽기 단위로 통상적으로 디스크 섹터의 배수가 된다.

디스크 배열은 다중의 디스크를 가진 디스크 서비스 시스템으로 디스크 액세스의 동시성과 병렬성을 지원하여 디스크 입출력의 성능을 향상시킬 수 있다. 동시성(concurrency)은 디스크 배열이 여러 개의 입출력 요구를 동시적으로 처리 가능한 성질을 말하고, 병렬성

(parallelism)은 하나의 디스크 입출력 요구를 서비스하기 위해 디스크 배열의 각 디스크들이 협조체제를 가지고 병렬적으로 동작하는 특성을 말한다

디스크 배열은 동기화 유무에 따라 동기화 디스크 배열과 비동기화된 디스크 배열. 이 둘의 장점을 결합한 혼합 구조로 분류할 수 있다[3, 4, 8]

동기화된 디스크(synchronized disk) 방법에서 모든 디스크들은 각 디스크의 헤드가 같은 장소에 위치하도록 동기화되어 모든 디스크들이 마치 하나의 큰 디스크를 형성하는 것처럼 동작한다. m 개의 디스크가 존재할 경우, 동기화 정도(ds : degree of synchronization)가 m 이면 완전히 동기화된 시스템이 된다. 이러한 기법은 읽어야 할 데이터에서 오버헤드가 차지하는 비중이 큰 경우에 유리하고 데이터가 비트 또는 바이트 단위로 인터리빙 될 수 있으며, 디스크 액세스의 병렬성 향상이 가능하다. 이 방식은 디스크들이 마치 하나의 큰 디스크처럼 동작을 하므로 과학응용과 같은 요구의 크기가 큰 경우에 적합하다. 그러나 동기화된 디스크는 별도의 동기화 제어장치가 필요하고 이 장치를 만들기 힘들다는 단점이 있다.

비동기화된 디스크 배열은 각 디스크가 독립적으로 회전하며 디스크 암도 독립적으로 동작하여 여러 개의 디스크 입출력 요구가 여러 디스크에 의해 동시 액세스되며 디스크 상에 데이터가 블록단위로 저장된다. 즉, $ds=1$ 인 경우로 디스크 배열을 구성하고 있는 m 개의 디스크들이 서로 독립적으로 동작을 하는 독립적인 드라이브(independent drives) 방식이다. I/O 요구가 디스크 배열 시스템의 전역 버퍼(global buffer)에 들어오고 요구되어진 I/O 요구를 서비스하기 위해 디스패처(dispatcher)는 해당 데이터가 존재하는 디스크에 요구를 보낸다. 각 디스크의 제어기는 지역 버퍼(local buffer)에 들어온 요구들을 주어진 순서(FIFO) 혹은 다른 스케줄링 정책에 의하여 처리하게 된다.

혼합 구조는 일정한 수의 디스크를 갖는 여러 개의 그룹으로 이루어져 있으며, 하나의 그룹 내에 있는 디스크들은 동기화되어 하나의 디스크가 움직이는 것처럼 동작을 하고 각 그룹들은 서로 독립적으로 움직인다. 이 구조에서는 동기화 정도가 같은 디스크 배열 혹은 동기화 정도가 다른 디스크 배열을 가질 수 있다. 예를 들어, m 개의 디스크들로 구성된 디스크 배열의 동기화 그룹수가 g 이면 동기화 정도는 $ds=m/g$ 가 된다.

3.3 성능 지수

모의실험에서 고려한 성능 지수는 크게 디스크 액세스 시간 및 최대 허용 스트림수이나 그 중 본 절에서는 최대 허용 스트림수에 대하여 기술하기로 한다.

단일 디스크에서 동시에 서비스해야 하는 클라이언트 수를 n 이라 하면 이러한 시스템은 식 (3)을 만족해야 한다.

$$\sum_{i=1}^n (T_{seek}^i + T_{rotation}^i + T_{transfer}^i) \leq \min(T_{playback}) \quad (3)$$

여기서 저장된 데이터가 비디오라는 점을 고려하고 Zero-latency read 기법을 가정하여 회전지연시간을 0으로 두며 탐색시간을 최대 값으로 두게 되면 \sum 항을 식 (4)와 같이 간단하게 만들 수 있다.

$$T_{max, seek} + n \times \frac{s}{D_{disk}} \leq T_{playback} \quad (4)$$

따라서 최대수용 스트림수 n 은 식 (5)에 따라 구해질 수 있다.

$$n \leq \frac{T_{playback} - T_{max, seek}}{s/D_{disk}} \quad (5)$$

디스크 배열에서의 최대수용 스트림수는 배열 구성 방법에 따라 조금씩 다르다. 우선 동기화된 디스크 배열의 경우에 전체 수용 가능한 스트림수가 n' 이고 동기화된 디스크수가 m 이면 식 (6)을 만족해야 한다.

$$T_{max, seek} + n' \times \frac{s}{D_{disk}} \leq m \times T_{playback} \quad (6)$$

여기서 n' 를 구하면 식 (7)과 같다.

$$n' \leq \frac{m \times T_{playback} - T_{max, seek}}{s/D_{disk}} \quad (7)$$

독립된 디스크 배열인 경우는 단일 디스크에서의 수용 스트림수의 m 배라고 생각하면 된다. 즉 조건식은 식 (8)과 같고 전체 스트림수는 n 의 m 배가 된다.

$$T_{max, seek} + n \times \frac{s}{D_{disk}} \leq T_{playback} \quad (8)$$

혼합구조인 경우는 그룹수(g)만큼은 비동기적인 디스크의 성질을 따르고 degree 수만큼은 배열의 동기적인 성질을 따른다 따라서 각 그룹에서의 최대수용 스트림수를 n'라 할 때 추기는 degree수만큼 증가하므로 $\frac{m}{g}$ 배 늘어난다. 그러므로 전체적인 스트림수는 $g \times n'$ 이다

$$T_{min,seek} + n' \times \frac{s}{D_{disk}} \leq \frac{m}{g} \times T_{flavact} \quad (9)$$

여기서 주의할 것은 $n' \geq g \times n \geq m \times n$ 이다 즉 동기화된 디스크 배열의 경우가 최대수용 스트림수 측면에서 가장 우수하다는 것이다

4. 디스크 스케줄링 기법

VOD 시스템의 스케줄링 기법에 대해서는 지금까지 많은 연구가 진행되어 왔다[9-11]

스케줄링이 디스크 블록을 대상으로 하며 물리적인 저장 위치는 파일 시스템에서 결정하기 때문에 비디오 서버는 통신망으로 연결된 고객으로부터 물리적 디스크까지 종결간 시간계약 조건이 부여되므로 기존의 인터페이스만으로는 이러한 시간 제약 조건을 부여하기가 힘들고 반드시 서버 작성자에게 스케줄링 기능을 제공해야 한다.

디스크 스케줄링은 디스크 서브시스템의 구조와 디스크 장치의 특성, 그리고 응용의 요구사항 등에 의해 다양한 방법들이 존재한다 그러한 방법들을 탐색시간, 회전지연 시간, 종료시한 등과 같은 스케줄링의 결정 요소에 따라 물리적 차원 스케줄링, 논리적 차원 스케줄링, 구조적 차원 스케줄링 등으로 <표 1>과 같이 구별할 수 있다[12].

<표 1> 스케줄링 요소에 따른 디스크 스케줄링 분류

요소 구분	스케줄링 결정 요소	스케줄링 분류
구조적 요소	디스크의 다중성, 중복성 (디스크 배열, 디스크 반시)	구조적 차원 스케줄링
논리적 요소	종료시한 도착시간	논리적 차원 스케줄링
물리적 요소	탐색시간 회전지연 시간	물리적 차원 스케줄링

물리적 차원 스케줄링은 탐색시간을 최적화하는 방

법, 회전지연시간을 최적화하는 방법 및 이들의 장점을 취한 복합방법 등이 있다. 탐색시간을 최적화하는 방법으로는 SSTF(Shortest Seek Time First)와 SCAN, C-SCAN, N-SCAN 등이 있으며, SSTF와 SCAN의 방법을 복합적으로 이용하는 복합형으로 V-SCAN이 있다. 회전지연시간을 최적화하는 방법에는 이동헤드를 가진 디스크에서의 스케줄링 방법 등이 있다.

논리적 차원의 스케줄링은 입출력 요구에 주어진 시간 조건을 기준으로 스케줄링하는 방법으로 도착시간을 기준으로 하는 방법과 종료시한을 기준으로 하는 방법이 있다. 도착시간을 기준으로 하는 방법으로는 FCFS가 있다. 종료시한을 기준으로 하는 스케줄링 방법은 낮은 실기율과 짧은 종료시한 초과시간 달성을 그 목표로 한다. EDF(Earliest Deadline First), SCAN-EDF, D-SCAN(Earliest Deadline SCAN) 및 FD-SCAN (Feasible Deadline SCAN) 등이 있다.

구조적 차원 스케줄링 방법 중, 반시형 디스크 서브시스템에서의 스케줄링은 종료시한 만족도를 향상시키기 위한 큐 스케줄링 방법들과 디스크 쓰기 지연으로 디스크 입출력의 성능을 향상시키는 방법들이 있다. 디스크 배열에서의 스케줄링에서는 실시간성을 만족하고자 할 때는 디스크 배열의 구조, 종료시한, 디스크 앞의 움직임이 함께 고려되어야 하며 디스크 배열을 이용한 비디오 입출력 응용을 지원하기 위해서는 비디오 등과 같은 스트림의 특성, 디스크의 구조, 디스크 앞의 움직임이 동시에 고려되어야 한다

[11]에서는 디스크 스케줄링 알고리즘을 사용되는 자원의 양에 따라 SCAN류, Round-Robin류, 그리고 GSS류로 분류하였다.

라운드 로빈 알고리즘(Round-Robin Algorithm)은 헤드의 위치나 실기율에 상관없이 고정된 순서로 서비스되는 방법이다 GSS(Group Swept Scheme)는 세상이 각 그룹의 끝에서 시작되므로 지연은 각 그룹의 길이로 제한되며 필요한 머프크기도 한 그룹 안에 있는 스트림수에 대해서만 추가적으로 필요하므로 스캔에 비해 비피요구량을 줄일 수 있다.

<표 2>는 이들 알고리즘들의 초기지연 및 비피요구량을 표로 나타낸 것이다. 위 분류법에서 논리적 차원 스케줄링은 거의 다 SCAN류에 속한다. 표에서도 알 수 있듯이 시작지연 및 비피요구량은 Round-Robin이 가장 효율적이다 하지만 이 방법은 부하가 작을 경우에는 다른 알고리즘보다 우수하지만 부하가 커질수록

성능이 많이 악화된다.

〈표 2〉 디스크 스케줄링 알고리즘의 초기지연 및 버퍼사용량

일고리증 비교항목	SCAN	GSS	Round-Robin
시작지연	2C	(1-1/g)C	C/n
소요버퍼량	2nU	(n+n/g)U	(n+1)U

그러나 이러한 분류방법들은 실제 VOD 시스템의 검색 기법과는 거리가 있다. VOD 시스템에서의 서비스 제공 방법은 각 스트림에 대해 라운드를 기반으로 해서 연속적으로 읽어내야 하는 구조를 사용하고 있다.

[9]에서는 이러한 특성을 고려하여 스케줄링 기법을 선택할 때 데이터 액세스 기법을 필수적으로 고려하도록 제안하고 있다. 데이터 액세스 기법으로는 CTL(Constant Time Length)과 CDL(Constant Data Length)이 있다[13-15]. CTL은 한 서버에 들어오는 요청들에 대해 한 개의 공통된 주기마다 필요한 양을 읽어내도록 하고 있다. 이때 SCAN류의 스케줄링은 테드라인이 동일한 것을 원칙으로 하므로 적합하겠지만 EDF는 테드라인에 따라 스케줄링 우선 순위를 결정하는 것이기 때문에 별 의미가 없다. 반면 CDL은 가변적인 시간에 대해 동일한 크기의 데이터를 연속적으로 읽어내도록 고려하고 있다. 그러므로 이 경우에는 전체적인 주기라는 개념이 희석되므로 SCAN류의 알고리즘은 적용될 수 없다. 그러므로 CTL 액세스 정책은 SCAN류의 스케줄링과 결합하고 CDL 액세스 정책은 EDF류와 결합시키는 것이 가장 바람직하다고 판단된다. SCAN류의 알고리즘에는 LOOK, GSS, RR 등이 있고 EDF류는 EDF나 SCAN-EDF같은 스케줄링 기법들이 있다.

두 정책을 비교해보면 CTL 정책이 탐색 최적화를 추구하는 알고리즘들과 결합되므로 CDL에 비해 최대 수용 스트림수에서 약간 우수하다고 할 수 있다. 그러나 이러한 CTL 기법은 읽는 데이터량이 주기마다 달라지므로 필요한 버퍼 공간은 이중버퍼를 사용할 경우 두 주기동안 읽어야 할 데이터 중 최대치에 맞추어 대역폭 및 버퍼 할당이 이루어져야 하므로 자원의 낭비가 있게 된다. 그러나 CDL 정책은 이러한 자원의 낭비가 없다는 장점이 있다.

VOD 시스템에서 데이터 액세스는 일반적으로 라운드 단위로 이루어진다. 그러므로 VOD 시스템에서 스

케줄링 정책을 적용할 때는 먼저 어떤 액세스 정책을 사용할 것인지 결정하여야 한다.

4.1 기존의 디스크 스케줄링 기법

디스크 접근(access)에서 탐색 시간(seek time) 및 회전 지연 시간(rotational latency)을 감소시키고, 높은 처리율을 얻고 각 스트림에 공정한 집근을 제공하기 위하여 기존의 디스크 스케줄링 알고리즘들 중 대표적인 것은 FCFS(First Come First Served), SSTF(Shortest Seek Time First), SCAN 등이 있다.

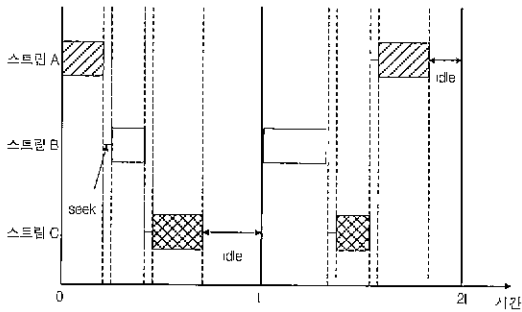
FCFS는 가장 먼저 도착한 데이터 요청을 가장 먼저 처리하는 방법이다. 이 방법은 쉽게 프로그램할 수 있고 본질적으로 공평성이 유지되는 특징을 지니고 있다. 그러나 경부하일 때는 별 문제가 없으나 부하의 증가와 함께 포화현상이 발생, 응답 시간(response time)이 증가하게 되는 단점이 있다. 데이터 요청이 디스크 상에 균일하게 분포되어 있으면 임의의 탐색형태가 되어 탐색 형태의 최적화에 아무런 기여를 하지 않는다.

SSTF는 근본적으로 SJF(Shortest Job First) 알고리즘의 형태로, 탐색시간을 감소시키기 위해 어떤 주어진 디스크 요청을 처리하기 위해서는 헤드가 먼 곳까지 이동하기 전에 현대 헤드 위치에 가까운 모든 데이터 요청을 먼저 처리하는 것이다. FCFS보다는 처리율 면에서 좋다. 실린더의 제일 안쪽과 바깥쪽에서 디스크 요청의 기아 현상이 발생할 수 있고, 적정 부하에서 평균 응답 시간이 높으며 응답 시간의 분산(variance)이 크다는 단점이 있다.

4.2 라운드 기반(round-based) 스케줄링

여기서 라운드는 동시에 서비스 받는 모든 스트림들에 공통되는 테드라인을 의미한다. (그림 2)에서 라운드 기반 스케줄링의 예를 보여주고 있다. 세 개의 스트림(A, B, C)이 동시에 서비스 받고 있으며 그들의 공통된 주기는 t 단위로 진행된다. 그러므로, A, B, C는 라운드마다 모두 같은 테드라인을 가진다고 할 수 있다. 이러한 스케줄링에서는 같은 테드라인을 가진다는 점을 이용하여 탐색시간을 최적화 함으로써 성능을 향상시킬 수 있다.

본 절에서는 몇 가지의 라운드 기반 스케줄링 기법에 대하여 기술하였다.



(그림 2) 라운드 기반 스케줄링의 예

4.2.1 SCAN 스케줄링 기법

SCAN 알고리즘에서는 일단 헤드가 한 방향으로 움직이기 시작하면 그 방향으로 계속 움직이면서 요청을 처리한다. 이러한 알고리즘을 엘리베이터 형(elevator-type) 알고리즘 혹은 단방향 알고리즘이라고 한다. SCAN에서는 움직이는 방향으로 더 요청이 있으면, 헤드를 디스크의 끝까지 이동한 후에 다시 방향을 바꾸는 방법이다 따라서 SCAN에서는 입출력 헤드가 디스크의 양쪽끝을 왕복하면서 요청된 트랙에 대한 처리를 해 나간다.

만약, 새로운 디스크 요청이 헤드의 바로 앞 트랙에 대한 것이면 즉시 처리되지만, 반대로 금방 지나온 헤드 바로 뒤에 있다면 헤드가 디스크의 끝까지 이동하고 나서 되돌아 올 때까지 기다려야 한다 그러므로, 헤드가 한쪽 끝에 이르러 방향을 바꾸어야 할 시점에서 요청 밀도가 높은 쪽은 최초의 시작 부분이며, 나중에 처리된 헤드 바로 뒷부분은 확률적으로 비교적 밀도가 낮다. 따라서, 밀도가 높은 쪽의 요청은 상당히 오랜 시간을 기다리게 된다.

SCAN에서 n 스트림을 검색하기 위하여 요구되는 버퍼 공간의 최소 크기는 $2nU_{scan}$ 이다 여기서 U_{scan} 은 스캔 스킴에서 n 스트림을 지원하기 위하여 필요한 최소 전달 단위이다

4.2.2 Round-robin 스케줄링 기법

각 라운드동안 비디오 서버는 각 스트림에 대하여 임의 길이(어떤 때는 없음)의 미디어 블록 순서를 검색한다[16] 연속 매체 검색의 주기적인 성질을 이용한다. 각 라운드마다 변하지 않는 고정된 순서로 스트림들을 서비스한다 지연을 본질적으로 감소시키는 데이터 배열 알고리즘이 라운드 로빈 스케줄링과 연계되어

때때로 사용된다 주 단점은 EDF처럼 라운드동안 검색되는 미디어 블록들의 상대적 위치를 이용하지 않는 것이다.

라운드 로빈에서 n 스트림을 검색하기 위하여 요구되는 버퍼 공간의 최소 크기는 $(n+1)U_{RR}$ 이다. 여기서 U_{RR} 은 라운드 로빈 스킴에서 n 스트림을 지원하기 위하여 필요한 최소 전달 단위이다. 라운드 로빈의 경우, 최악의 집진 시간은 디스크 안이 가장 바깥쪽에서 가장 안속으로 탐색을 하고, 완전히 회전을 하고, 안쪽으로부터 데이터를 읽을 때 발생한다.

4.2.3 GSS(Grouped Sweeping Scheme) 스케줄링 기법

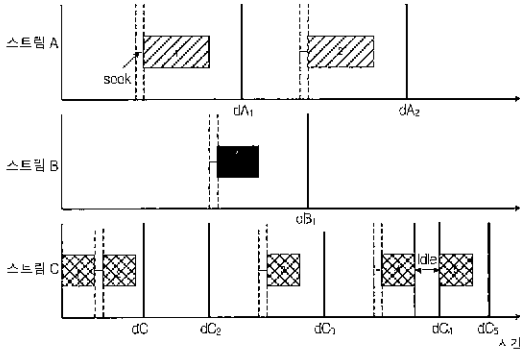
각 라운드를 비슷한 데드라인을 가지고 있는 여러 개의 그룹으로 분할한다[11]. 각 스트림은 하나의 그룹에 배정되며 그룹들은 각 라운드에서 고정된 순서로 서비스된다. 각 그룹 내에서는 SCAN 디스크 스케줄링 알고리즘이 사용된다

만약 모든 스트림들이 동일한 그룹에 배정된다면 GSS는 SCAN으로 된다. 반면에 각 스트림이 자기 자신의 고유한 그룹에 배정된다면 GSS는 라운드 로빈으로 된다 최적의 그룹 수를 유도함으로써 서버는 연속적인 스트림 검색 지연에 대하여 라운드 길이 감소의 균형을 취할 수 있다.

4.3 데드라인 기반(deadline-based) 스케줄링

데드라인 기반 스케줄링은 특히 멀티미디어 데이터가 기변 비트율(VBR-variable bit rate)적인 트래픽 특성을 가지므로 이로 인해 각 블록의 데드라인이 다른 점을 이용하기 위한 것이다. 그러므로 이 스케줄링을 이용하기 위해서는 데드라인이 모두 다르도록 조정할 수 있으며 VBR적인 트래픽 특성을 고려하여 각 주기마다 고정된 양의 데이터를 읽음으로써 데드라인을 다르게 조정할 수 있다 (그림 3)에서는 스트림 A, B, C가 각각 고정된 양의 데이터를 다른 데드라인 안에 읽어내는 과정을 보여주고 있으며 이중 버퍼를 사용한다고 가정한다 여기서 C가 데드라인이 가장 작으므로 가장 먼저 블록 1과 블록 2를 읽은 후 A에 대한 블록을 읽는다. 그리고 나서 데드라인을 비교하여보고 데드라인이 가장 작은 B에 대한 블록을 읽는다. C에 대한 블록 4를 읽고는 휴지시간(idle time)이 생기는데 이것은 시스템이 각 스트림에 대해 이중 버퍼를 사용

한다고 가정했기 때문이다.



(그림 3) 데드라인 기반 스케줄링의 예

4.3.1 EDF(Earliest Deadline First) 스케줄링 기법

이 알고리즘은 데드라인을 가진 테스크의 실시간 스케줄링에 대하여 가장 잘 알려진 알고리즘이다[16]. 가장 빠른 데드라인을 가진 미디어 블록을 검색하기 위하여 계획한다. 이 알고리즘은 과도한 탐색 시간과 회전 지연 시간을 야기시킬 수 있고 낮은 서버 이용율이 기대된다.

4.3.2 SCAN-EDF 스케줄링 기법

이 알고리즘은 탐색 시간의 최적화를 위한 SCAN 알고리즘과 실시간 보장을 위한 EDF 알고리즘의 장점을 결합한 것이다[17]. EDF에서와 같이 가장 빠른 데드라인을 가진 요구가 제일 먼저 서비스된다. 그러나 여러 개의 요청이 동일한 데드라인을 가진 경우 각각의 블록들은 SCAN 알고리즘으로 접근된다. 알고리즘의 효과는 동일한 데드라인을 가지는 요청의 수에 의존한다. 연속 매체는 본질적으로 요청을 배치(batch)로 하지 않기 때문에, 이 알고리즘은 연속 매체에 고유한 디스크 스케줄링 알고리즘이다. 미디어 블록 검색 데드라인이 배치(batch)될 때 SCAN-EDF는 단지 SCAN으로 된다

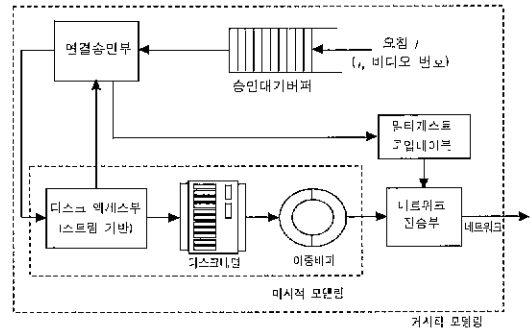
SCAN-EDF는 EDF 알고리즘을 약간 수정하면 쉽게 구현될 수 있다 D_i 를 테스크 i 의 데드라인이라 하고, N_i 를 트랙 위치라 하면 데드라인이 식 (10)과 같은 조건을 만족하는 $D_i + f(N_i)$ 가 되도록 수정할 수 있다.

$$D_i + f(N_i) \leq D_j + f(N_j) \quad \forall \quad D_i \leq D_j \quad (10)$$

5. 모의실험 및 성능 평가

5.1 모의실험 시스템 구성

전체적인 시뮬레이터 구조는 (그림 4)와 같다



(그림 4) 시뮬레이터 구조

(그림 4)에서 보는 바와 같이 미시적 모델링은 저장소의 액세스 성능과 관련한 사항만을 고려하였다. 즉 버퍼량과 네트워크 대역폭은 충분한 것으로 가정하고 액세스와 관련된 성능분석을 수행하기 위해 요청도착의 수는 모델을 사용하여 구성한 것이 아니라 모의실험 초기에 입력 파라미터로 결정하여 프레임 손실이 발생하지 않는 범위의 상한 값들을 구하였다.

디스크에 대한 접근방법으로는 파일 시스템 기반(file system-based) 접근법과 스트림 기반(stream-based) 접근법이 있다 파일 시스템 기반 접근법은 요청-서비스를 반복하는 구조를 말하며, 스트림 기반 접근법은 한번의 요청 후에 연속적인 주기별 서비스가 이루어지는 구조를 말한다 본 논문에서는 시스템이 스트림 기반 접근법을 사용한다고 고려하여 한번 승인된 요청에 대한 스트림은 서비스가 끝날 때까지 다시 요청이 없더라도 주기적인 서비스를 수행하는 것으로 가정하였다.

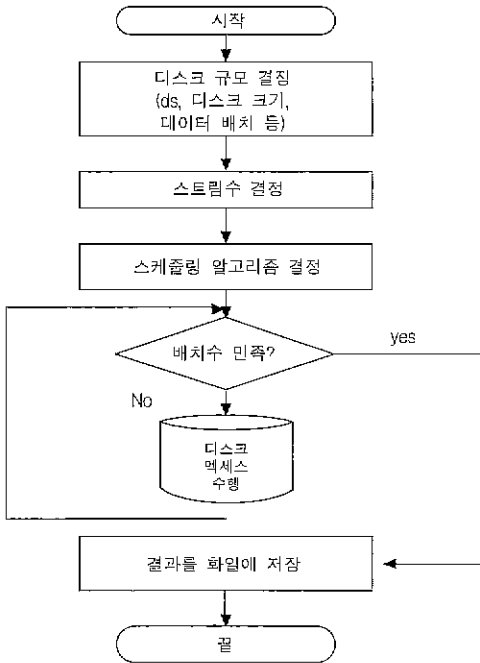
미시적 모델링에서는 디스크의 최대 성능을 평가하는 것이 그 목적이므로 스트림수를 모의실험 초기에 고정시키는 반면, 거시적 모델링에서는 가변 스트림수를 기본 원칙으로 하였다. [13]에서 제안된 CTL, CDL의 의미는 한 요청 스트림에 대한 개념으로 CBR일 때 이종 환경에 적용된 CTL과 CDL은 원래 제안된 의미와 일치하는 것은 아니다. 그러나 CBR 트랙의 특성에서는 동시스트림에 대한 서비스에서 CTL은 모든 스트림을 한 라운드로 묶는 것이 가장 바람직한 것으로 고

리되며 CDL의 경우도 모든 스트림에 대해 한 라운드에서 읽는 크기가 같다고 가정하는 것이 바람직한 것으로 고려되므로 CBR 또는 VBR에 대한 언급없이 쓰기로 한다[14].

5.2 미시적 모델링에 대한 성능분석

5.2.1 모의실험 흐름도

미시적 모델링을 위한 모의실험의 흐름도는 (그림 5)와 같다. 모의실험 결과의 분석을 위하여 배치 평균법(batch mean method)을 이용하였다.



(그림 5) 미시적 모델링을 위한 모의실험 흐름도

5.2.2 모의실험을 위한 가정

본 절에서는 디스크 배열 구조를 대상으로 한 미시적 모델링에 대한 모의실험 가정을 기술하고자 한다. 실험들은 스케줄링 정책 및 디스크 동기화 구조에 따른 최대수용 스트림수 및 디스크 전달단위의 경향 등에 초점을 맞추었다.

디스크 규모는 한 섹터는 512바이트, 한 트랙은 80개의 섹터로 구성되고 한 실린더는 20개의 트랙으로 이루어져 있으며 총 2732개의 실린더로 한 개의 단일 디스크를 구성한다. 그러므로 한 디스크의 용량은

2.098GB 정도의 용량을 가지는 것으로 하나의 비디오 내용을 수용하기에 충분한 용량이라 가정하고 기본적으로는 하나의 디스크가 하나의 비디오 용량을 가지는 것으로 고리한다. 또한 디스크 배열을 구성하는 맴비 디스크의 수는 16개를 가정하였으며 이를 바탕으로 ds 를 고려하여 성능을 평가한다.

본 논문에서의 실험을 위한 기본적인 가정은 다음과 같다

- ① 읽기 연신만 고려
- ② 재생률 동일(1.5 혹은 3Mbps)
- ③ 연속 블록 배치기법 사용
- ④ 한 영화의 크기는 한 디스크 크기로 고정
- ⑤ 블록의 크기는 트랙의 정수배수가 되도록 결정
- ⑥ 네트워크는 충분한 대역폭을 가지므로 전체 서버 성능은 저장시스템 성능에 의해 좌우

블록 배치 방법은 기본적으로 연속배치 방법을 사용하였다. 단일디스크인 경우는 한 디스크 안에 한 비디오의 내용이 정확하게 들어가 있다고 가정하고 최대수용 스트림수 및 액세스 시간만을 고려하였다. 디스크 배열인 경우는 배열 구성방법에 따라 독립적인 디스크, 동기화된 디스크로 나누어 각각 단일디스크와 같은 블록배치 방법과 스트라이핑 기법을 사용하는 방법 등을 적용하였다[4].

5.2.3 디스크 배열의 성능분석

가. 탐색시간 모델의 영향

탐색모델의 사용이 최대수용 스트림수에 미치는 영향을 분석하기 위해 LOOK 스케줄링 알고리즘을 사용하여 단일디스크에 대하여 모의실험을 수행하였다 <표 3> 탐색시간 모델별 최대수용 스트림수에는 변화가 없었고 읽어야 할 최소디스크 전달단위는 비선형 모델이 약간 우수한 것으로 나타났다. 비선형 모델이 우수하게 나타난 이유는 한 디스크에서 최대로 수용할 수 있는 스트림수가 5~6개이므로 평균 스트림간 트랙거리가 500 트랙 정도가 되어 선형모델보다 탐색시간이 작게 질리기 때문이다.

탐색시간모델을 디스크 배열에 적용시켰을 때 최대수용 스트림수 측면에서 비선형 탐색모델의 경우 선형 탐색모델과 차이가 없음을 확인할 수 있었다. 이것은 최대수용 스트림의 경우 이동 트랙간의 평균지연이 모델에 따라서 별 차이가 없기 때문인 것으로 분석된다

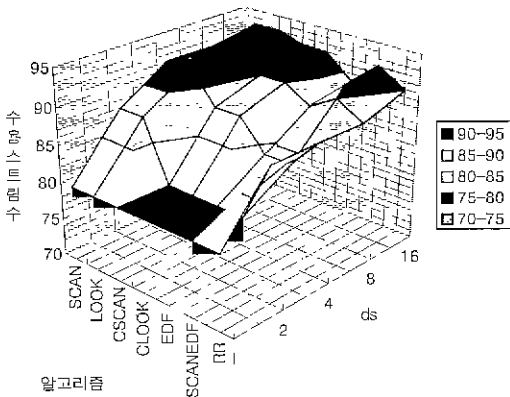
<표 3> 동일한 스트림수의 수용을 위한 선형 및 비선형 모델의 최소 디스크 전달단위(KB) 비교

스트림수 선형모델	1	2	3	4	5
선형	40	80	80	120	210
비선형	40	40	40	80	160

나. 스케줄링 알고리즘의 영향

디스크 배열의 성능은 물리적인 제한 요소 외에 멤버 디스크수, 동기화 정도, 블록크기 및 스케줄링 정책에 영향을 받는다. 이들 요소들은 서로 상관성을 가지므로 본 논문에서는 멤버 디스크수를 16으로 가정하여 스케줄링 정책의 영향을 기준으로 ds 및 블록 크기의 영향 등을 살펴보고자 한다

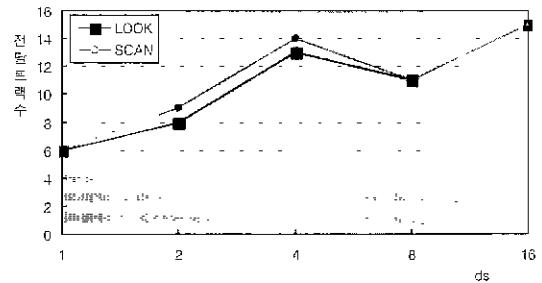
먼저 최적의 스케줄링 알고리즘을 선택하기 위해 LOOK, SCAN, EDF, SCAN-EDF, RR 알고리즘을 적용하였을 때의 ds 별 최대수용 스트림수를 비교하여 (그림 6)에 나타내었다



(그림 6) 스케줄링 정책별 최대수용 스트림수 비교

(그림 6)에서의 결과를 비교해 보면 동일한 스케줄링 알고리즘을 적용한 경우, ds 값이 증가함에 따라 최대수용 스트림수가 증가하고, 스케줄링 알고리즘 측면에서 본다면 SCAN 및 LOOK이 가장 우수함을 확인할 수 있다. 또한 RR은 동기화 정도가 낮을 경우에는 EDF나 SCAN-EDF보다 우수함을 확인할 수 있다. 스케줄링 알고리즘의 영향은 단일디스크에서보다는 디스크 배열에서 수용 스트림수가 증가하기 때문에 그 영향이 커짐을 확인할 수 있다. (그림 6)에서 LOOK과 SCAN이 가장 우수한 성능을 보여주고 있다.

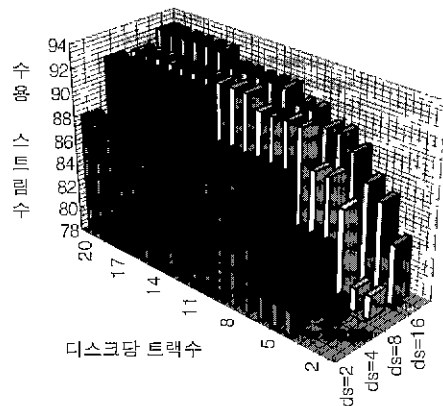
이 두 알고리즘들을 비교하기 위해 최대 스트림수를 수용할 수 있는 최소 디스크 전달단위크기를 비교하였다 (그림 7)에서는 LOOK이 SCAN보다 최소 디스크 전달단위 즉 최소 소요 버퍼 측면에서 약간 우수함을 확인할 수 있다



(그림 7) ds에 따른 디스크별 최소전달단위(트랙갯수) 비교(LOOK vs. SCAN)

위 결과들에 바탕을 둔다면 스케줄링 알고리즘은 가장 우수한 성능을 보여주는 LOOK으로 고정시킬 수 있다.

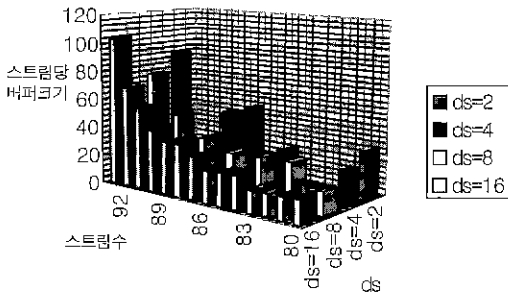
동기화 정도(ds)에 따른 최대수용 스트림수를 모의 실험으로 얻은 결과는 그림 8과 같다. 그림에서도 확인할 수 있듯이 동기화가 커질수록 최대 수용스트림수도 높아지는 경향을 가진다. ds가 16인 경우 최대 93 스트림까지 수용할 수 있음을 확인할 수 있다



(그림 8) LOOK 적용시 디스크 배열의 최대 수용 스트림수

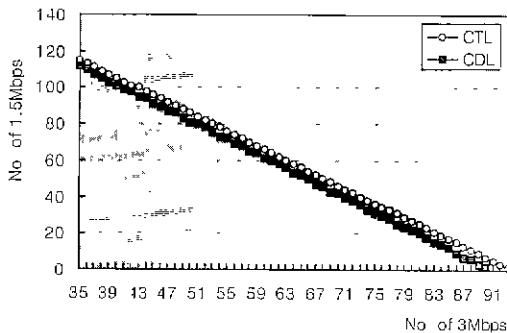
(그림 9)는 스트림당 소요버퍼 영역 측면에서 산출한 결과를 보여준다. 동기화가 커질수록 스트림당 소

요 버퍼가 작게 필요하다는 것을 확인할 수 있다. 예로서, 92 스트림인 경우를 보면, 동기화가 16인 경우가 소비버퍼 크기가 제일 작기 때문에 가장 효과적임을 알 수 있다



(그림 9) LOOK 적용시 동기화 정도에 따른 소비버퍼크기

(그림 10)은 CTL과 CDL 데이터 액세스 정책에 대하여 3Mbps와 1.5Mbps 데이터 전송률인 경우 최대 허용 스트림수에 대한 스케줄러블(schedulable) 영역을 표시한 것이다. 디스크 전달단위는 디스크별로 한 실린더 크기가 되도록 하였다. CTL이 최대 허용 스트림 수 측면에서 약간 우수(이유는 LOOK 스케줄링을 사용하므로 탐색 오버헤드에서 이득이 있음)한 것을 보여준다.



(그림 10) CTL과 CDL 데이터 액세스 정책의 최대 허용 스트림수 비교

5.2.4 타당성 검토

단일디스크인 경우 최대수용 스트림 수는 스케줄링 같은 논리적 요소들보다는 오히려 디스크 시스템 자체의 고유한 물리적인 성능에 크게 영향을 받는 것으로 판단된다. 그 결과의 타당성은 다음과 같이 증명할 수

있다.

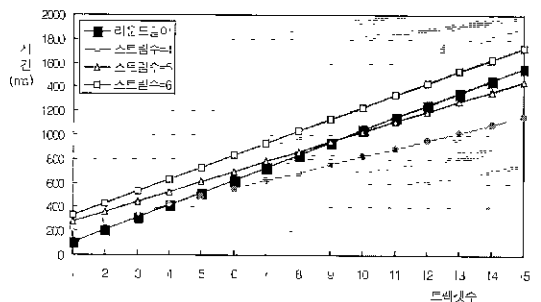
예를 들어, 재생률($R_{playback}$)을 3Mbps라 두고 디스크 회전속도(rpm)는 3600rpm, 한 트랙의 용량($size_{track}$)은 40KB라 하면, 1초에 단일 디스크로부터 읽을 수 있는 데이터량(D_{disk})은 2400KB/sec이며 재생을 위해 1초안에 읽어야 할 데이터량($D_{playback} = \frac{R_{playback}}{T_{playback}}$)은 3Mbits이다. 최대 탐색시간(max_{seek})은 38ms, 헤드 스위치 시간은 0.7ms라 하고 서비스해야 할 스트림 수를 n 이라 하면, n 은 식 (11)을 만족시켜야 한다.

$$\frac{1000 \times D_{track} \times n}{D_{disk}} + max_{seek} \times n \quad (11)$$

$$+ T_{head-switch} \times n < 1000 (ms)$$

$$\therefore n < 5.03이다.$$

그러므로 이러한 조건에서의 최대 수용 스트림수는 5이다. (그림 11)은 블록 크기별 라운드 길이 및 스트림 수 n 의 크기에 따른 라운드의 길이를 나타낸다. 라운드 길이가 길어짐에 따라 스트림 수 4, 5는 일정한 시점에서부터 수용이 가능하지만 스트림 수 6에 대한 최소요구 라운드 길이는 스트림 검색을 위한 기본적인 라운드 길이보다 길어서 수용이 불가능하다는 것을 알 수 있으며 이것은 위의 계산 결과의 타당성을 입증한다.



(그림 11) 블록 크기별 라운드 길이 및 스트림수에 따른 라운드길이($R_{playback} = 3Mbps$)

최대수용 스트림수의 상한 값의 일반식을 구하기 위해 다음을 고려해보자 위의 식에서 탐색 및 헤드 스위치 시간과 관계되는 항은 상수로 고려하고 단순히 데이터 전송시간만 고려해보자 이때 전체부하에서 오버헤드가 차지하는 비율은 [18]에서와 같이 약 10%로

고려한다. 그러면 다음과 같은 식을 만들 수 있다.

$$\frac{D_{disk}}{n} \leq R_{playback} \quad (n \geq 1, \text{ 정수}) \quad (12)$$

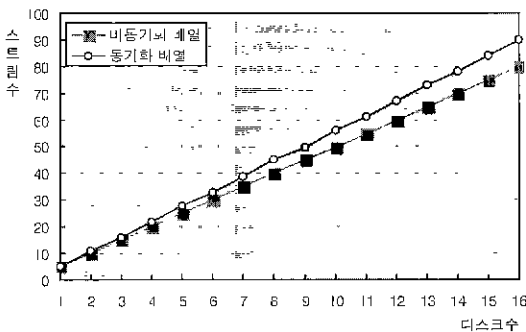
$$\frac{D_{disk}}{R_{playback}} \geq n \quad (13)$$

$R_{playback}$ 을 일정하다고 가정하면 n 은 D_{disk} 에 비례하는 관계를 나타낸다. 이러한 D_{disk} 는 디스크의 물리적인 등각에 의존하는 값으로 디스크 회전속도가 3600 rpm이고 트랙당 40KB를 읽을 수 있다면 D_{disk} 는 2400 KB/sec이므로 $6.25 \geq n$ 이다. 여기서 오버헤드가 차지하는 비율(10%)을 고려한다면 $6.25 \times 0.9 = 5.625$ 로 n 의 상한선은 5가 된다. 만약 디스크 회전속도를 7200 rpm이라 둔다면 최대수용 스트리프수는 $11 (\leq 11.25)$ 이 된다.

마지막으로 디스크 배열로 확장시켰을 때의 성능 검토를 위하여 단일 디스크의 경우를 다시 고려해보자. [18]에서는 디스크당 최대 수용 스트리프 수를 계산하기 위해 식 (14)를 사용하였다. 여기서 10%의 헤드룸을 고려하였다. 이 식을 통해 구한 단일 디스크에 대한 최대 수용 스트리프 수(n)는 5.625 즉 5명까지 최대로 수용 가능하다는 결론을 얻을 수 있다.

$$n = 0.9 \times \frac{D_{disk}}{R_{playback}} \quad (14)$$

이 식을 디스크 배열에 적용시켜 계산하여 보았다. 그 결과 (그림 12)에서 볼 수 있는 것처럼 디스크 배열을 구성하는 멤버 디스크 수가 늘어날수록 수용 스트리프수 측면에서 이득이 있다는 것을 확인할 수 있다.



(그림 12) 디스크수 별 최대수용 스트리프수의 계산치 비교

즉, 비동기화된 경우 디스크수가 16인 경우의 최대 수용 스트리프수는 80이 되지만, 동기화시킨 경우는 90 스트리프까지 수용 가능하다는 것을 확인할 수 있다. 위에서 직용한 최대 수용 스트리프수의 계산은 VOD 서버 설계 시 저장 시스템의 기본적인 크기를 결정하는데 사용될 수 있을 것으로 판단된다.

6. 결 론

비디오 저장소의 성능을 연구하기 위해서는 크게 두 가지 관점을 고려할 수 있다. 첫째는, 비디오 저장소의 성능이 저장소의 물리적인 특성에 의해 상한 값이 결정되므로 이러한 한계를 극복하기 위한 여러 대안들을 사용하는 것이다. 둘째는, 이러한 물리적인 특성에 의한 상한 값들을 바탕으로 사용 가능한 여러 기법들의 효율성을 최대화하는 것이다.

본 논문에서는 VOD 시스템의 저장소로 사용되는 디스크 배열의 성능을 비교분석하기 위하여 모의실험을 수행하였다. 성능분석을 통하여 단일디스크의 성능은 디스크의 물리적인 성능에 의해 주로 의존되는 것으로 발견되었다. 탐색시간의 정도와는 관계없이 일정 스트리프로 제한되는 것을 확인할 수 있었으며, 스케줄링 정책별에도 단일디스크인 경우 수용 스트리프수 측면에서는 큰 차이가 없음을 확인할 수 있었다.

이것을 디스크 배열로 확장했을 경우 스케줄링에 의한 성능의 영향이 커짐을 알 수 있었다. 디스크 배열은 단일 디스크의 물리적인 제한을 극복하고자 제안되어 여러 개의 단일디스크들이 하나의 디스크처럼 동작한다. 그러므로 비동기화된 디스크인 경우에는 단일디스크의 성능을 디스크수 만큼 확장한 것과 같으며, 혼합 구조나 비동기화 구조보다는 동기화된 디스크 구조가 최대 수용 스트리프수에서 더 우수함을 확인할 수 있었다. 회전 지연 오버헤드도 디스크 배열에서는 중요한 요소로 동기화된 구조에서도 스트라이핑 단위가 트랙크기의 배수가 될 때 성능이 최적임을 확인할 수 있었다.

수용 스트리프수에 대한 성능평가를 수행하였을 때 스케줄링 정책별로는 LOOK과 SCAN이 다른 알고리즘보다 효과가 있었으나 LOOK이 SCAN보다는 약간 더 우수하였다. 동기화 정도별로는 동기화가 커질수록 이득이 있었다. 그러나, 본 논문에서는 디스크 시스템 자체만을 고려하였기에 그러한 결과를 얻을 수 있었지만

이것을 네트워크와 인질시킵 경우는 우선 SCSI 용량도 고려해야 하고 네트워크 대역폭 및 비퍼 크기와 같은 자원량도 고려해야 하기 때문에 무한정 동기화 정도를 높일 수는 없을 것으로 판단된다.

VOD 시스템에서 저장 데이터는 실시간성 및 대용량성을 요구하므로 데이터 검색은 라운드클로 필요한 양의 데이터를 읽어내는 방법으로 이루어진다. MPEG으로 저장된 데이터는 VBR적인 데이터 특성을 가지며 VBR로 읽을 때의 액세스 정책으로는 CDL 또는 CTL로 읽을 수 있으며 각각은 장점 및 단점을 가지게 된다. 이러한 액세스 정책에 따른 디스크 스케줄링 정책이 이용되는데 본 논문에서는 그 관계를 규명하고자 하였다.

앞으로 본 논문에서 기술한 CTL과 CDL 액세스 정책을 이용하여 네트워크와 연결된 VOD 서버 시스템을 고려하여, 보다 실제적인 MPEG 비디오의 저장 환경에서 비디오 검색 기법과 전송 기법을 연계한 방법에 대한 성능 평가 작업을 수행하는 것이 향후과제이다

참 고 문 헌

- [1] D James Gemmell et al. "Multimedia Storage Servers - A Tutorial," IEEE Computer, pp.40-49, May 1995.
- [2] Guojun Lu, Communication and Computing for Distributed Multimedia Systems, Artech House, 1996.
- [3] 전용희, 박정숙, 박영덕, "VOD 시스템을 위한 버퍼링 기법", 한국통신학회지, 제14권 제7호, pp.156-166, 1997년 7월
- [4] 전용희, 박영덕, "VOD 시스템을 위한 데이터 저장 기법", 한국통신학회지, 제14권 제7호, pp.141-155, 1997년 7월
- [5] Tsun-Ping J. To and Babak Hamidzadeh. "Dynamic real-time scheduling strategies for interactive continuous media servers," Multimedia Systems Vol.7, pp.91-106, 1999.
- [6] 박정숙, 전용희, 박영덕, "VOD 시스템을 위한 디스크 배열의 성능 분석", 한국통신학회 추계종합학술대회 논문집, 제16권 제2호, pp.1308-1312. 1997. 11.
- [7] Chris Rucmmler and John Wilkes, "An Introduction to Disk Drive Modeling," IEEE Computer, pp.17-28, March 1994.
- [8] 박정숙, 전용희, "VOD 서버의 성능평가에 대한 연구", 한국통신학회 하계종합학술발표회 논문집, pp.9-10-944, 1998. 7
- [9] Sambit Sahu, Zhu-Li Zhang, Jun Kurose and Don Towsley, "On the Efficient Retricval of VBR Video in a Multimedia Server," Proc of MCS'97, pp.46-53, 1997.
- [10] 전용희, 박영덕, "VOD 시스템에서의 디스크 스케줄링 기법에 대한 연구", 정보통신합동 학술대회 논문집 (JCCI'97), pp.1378-1383, 1997.
- [11] Philip S Yu, Mon-Song Chen, and Dilip D Kaundlur, "Grouped sweeping scheduling for DASD-based multimedia storage management," Multimedia Systems 1, pp.99-109, 1993.
- [12] 황기태, 고성능 디스크 입출력을 위한 디스크 스케줄링 알고리즘, 서울대학교 대학원 컴퓨터공학과 공학박사 학위논문, 1994. 2.
- [13] Ed Chang and Avideh Zakhor, "Cost Analyses for VBR Video Servers," IEEE Multimedia, Winter '96, pp.56-71. 1996
- [14] 박정숙, 전용희, 박종훈, "VOD 서비스를 위한 데이터 액세스 정책의 성능평가", JCCI'98, pp.197-201, 1998. 4
- [15] Yong-Hee Jeon, Jung-Sook Park, "Performance Evaluation of Data Access and Retrieval Schemes for VOD Services." Proc. of ITC-CSCC'98, pp. 567-570, July 1998.
- [16] Ralf Steinmetz and Klara Nahrstedt, Multimedia - Computing, Communication & Applications. Prentice-Hall(Scies in Innovative Technology), 1995.
- [17] A L. Narashuma Reddy and Jim Wvllie, "Disk scheduling in a multimedia I/O system." ACM Multimedia'93, pp.225-233, June 1993.
- [18] Al Kovalick. The Fundamental Concepts of Vidco Servers. Hewlett Packard. 1994.



진 용 희

e-mail : yhjeon@cuth.cataegu.ac.kr

1978년 고려대학교 전기공학과
(공학사)

1985년~1987년 미국 플로리다공대
대학원 컴퓨터공학과

1989년 미국 노스캐롤라이나주립대
Elec. and Comp. Eng. 석사
(MS)

1992년 미국 노스캐롤라이나주립대 Elec. and Comp.
Eng. 박사(Ph D)

1978년~1978년 삼성중공업(주) 근무

1978년~1985년 한국전력기술(주) 근무

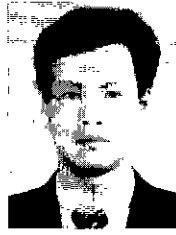
1989년~1989년 미국 노스캐롤라이나주립대 Elec. and
Comp. Eng. TA

1989년~1992년 미국 노스캐롤라이나주립대 부설 CCSP
(Center for Comm. & Signal Processing) RA.

1992년~1994년 한국전자통신연구소 통신시스템연구단
전임연구원

1994년~현재 대구가톨릭대학교 컴퓨터정보통신공학부
부교수

관심분야 : 차세대 인터넷, 멀티미디어 통신, 통신망
성능분석, 초고속망 프로토콜



이 상 학

e-mail : delta@med.yu.ac.kr

1987년 경일대학교 전자계산학과
(학사)

1993년 계명대학교 교육대학원
(석사)

1996년 대구가톨릭대학교대학원
전산통계학과 박사과정
수료

1983년~현재 영남대학교 의료원 생체역공학과 재직

관심분야 : PACS(Picture Archiving & Communication
System), 멀티미디어통신