



Phased RGSS : 연속매체 인출에 사용되는 디스크배열을 위한 개선된 디스크 스케줄링 방법

저자
(Authors) 이한욱, 김대웅, 박찬익

출처
(Source) [한국통신학회 학술대회논문집](#) , 1997.1, 232-235 (4 pages)
[Proceedings of Symposium of the Korean Institute of communications and Information Sciences](#) , 1997.1, 232-235 (4 pages)

발행처
(Publisher) [한국통신학회](#)
Korea Institute Of Communication Sciences

URL <http://www.dbpia.co.kr/Article/NODE00210824>

APA Style 이한욱, 김대웅, 박찬익 (1997). Phased RGSS : 연속매체 인출에 사용되는 디스크배열을 위한 개선된 디스크 스케줄링 방법. 한국통신학회 학술대회논문집, 232-235.

이용정보
(Accessed) 포항공과대학교
141.223.121.100
2016/05/09 17:01 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다.

이 자료를 원저작자와의 협의 없이 무단게재 할 경우, 저작권법 및 관련법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

The copyright of all works provided by DBpia belongs to the original author(s). Nurimedia is not responsible for contents of each work. Nor does it guarantee the contents.

You might take civil and criminal liabilities according to copyright and other relevant laws if you publish the contents without consultation with the original author(s).

Phased RGSS: 연속매체 인출에 사용되는 디스크배열을 위한 개선된 디스크 스케줄링 방법

이 한욱 김대웅 박찬익
포항공과대학교 전자계산학과, 포항공과대학교 전자계산학과 교수
(Han-Wook Lee) (Dae-Woong Kim) (Chan-Ik Park)

이 논문은 1995년도 한국학술진흥재단의 대학부설 연구소 연구과제 연구비, 1996년 한국과학재단 핵심전문연구과제 연구비, 그리고 1996년 정보통신연구관리단 대학기초연구과제 연구비에 의하여 연구되었음.

요약

VOD와 같은 멀티미디어 시스템은 많은 스트림의 요구에 대해 동시에 서비스를 만족할 수 있어야 한다. 또한 새로운 서비스의 요청이 들어왔을 때 처음으로 응답하는 시간이 가능한 작아지도록 유지하여야 한다.

본 논문에서는 연속매체 인출에 있어 좋은 성능을 보이는 GSS 방법을 개선시킨 RGSS 방법에서, 새로운 스트림이 도착할 때 초기 반응 시간이 커지는 문제점을 해결하였다. GSS에 대해 버퍼 요구량을 줄인 RGSS의 장점을 유지시키면서 새로운 스트림에 대한 초기 반응 시간을 크게 감소시키는 Phased RGSS 방법을 제안한다.

1. 서론

최근 들어 초고속 네트워크 기술과 저장 장치에 대한 기술의 발달은 VOD와 같은 새로운 형태의 멀티미디어 서비스를 가능하게 하였다. VOD와 같은 멀티미디어 시스템은 비디오 파일의 인출에 있어 deadline의 반응시간을 요구하는 일종의 실시간 비디오 멀티미디어 시스템으로 볼 수 있다. 이는 실시간 멀티미디어 서버가 전체 시스템의 요구뿐 아니라, 각각의 비디오 스트림들의 요구도 만족시켜야 한다는 것을 의미한다. 하지만 연속 매체 데이터의 성공적인 인출을 위해서는 이러한 각 스트림에 대해 반응시간을 보장해야 하는 실시간 조건뿐 아니라 인출에 필요한 버퍼의 크기를 줄이고 초기 반응 시간의 조건도 만족시켜야 한다.

버퍼의 크기를 최소화시키는 것은 동시에 서비스할 수 있는 스트림의 수를 최대화시키는 것과 일맥 상통한다. 버퍼의 크기가 무한하고 네트워크 속도의 제한만 없다면 동시에 서비스할 수 있는 스트림의 수도 무한히 증가하게 된다. 따라서 제한된 버퍼의 크기에서 스트림의 수를 최소화하려는 노력은 동시에 서비스 할 수 있는 스트림의 수를 최대화하는 것으로 볼 수 있다. 연속매체 인출을 위한 디스크 스케줄링 방법에는 대표적으로 SCAN-EDF[1]과 GSS[2, 3] 등의 방법들이 있다. 이외에도 GSS 방법을 RAID 디스크배열에 적용한 StreamingRAID [4] 방법이 있다. 그 중에서도 GSS 스케줄링 방법[3]은 '그룹'이라는 개념을 도입하여 일반적인 고정된 순서에 의한 방법[5]과 SCAN[6] 방법을 통합한 방법으로, 특히 연속매체 인출에 대한 응용에 좋은 성능을 보이고 있다.

최근에는 GSS 방법을 더욱 향상시킨 RGSS 방법[7]이 제안되었다. 이것은 디스크 공간을 일정한 영역으로 나누고, 데이터를 일정한 규칙에 따라 각 영역에 배치하는 데이터 배치방법을 제안하고, 그 각 영역에 대해 GSS 스케줄링 방법을 적용시킨 방법이다. 이렇게 함으로써 전체 디스크 공간상에서 동작되는 GSS 방법에 비해 RGSS 방법에서는 한 영역 내에서의 동작으로 축소되므로, GSS 방법보다 버퍼 요구량을 줄일 수 있다. 그러나, RGSS방법은 초기 반응시간이 영역의 개수에 따라 크게 증가하는 문제점을 가지고 있다.

초기 반응 시간이란 클라이언트가 비디오 서버에 서비스를 요청한 후, 첫 번째 비디오 프레임의 데이터가 도착할 때까지 걸리는 시간을 말한다. FIFO 방법이나 SCAN, GSS 방법들은 새로운 스트림의 요구가 들어오면 적어도 다음 서비스 주기(즉, 모든 스트림들에 대해 한번 처리하는데 걸리는 시간)에 이 새로운 스트림을 처리할 수 있다. 하지만 데이터

를 영역에 기반하여 배치할 경우 디스크 헤드가 새로이 도착한 스트림의 첫 번째 프레임이 들어 있는 영역에 도달할 때까지 서비스가 보류되게 된다. 따라서 영역에 기반한 GSS가 비록 버퍼 요구량을 줄여 주는 것이기는 하지만 어느 초기 반응 시간과의 trade off로 성립한다는 것을 알 수 있다.

본 연구에서는 영역에 기반한 RGSS 방법과 동일한 버퍼 크기를 요구 하면서 초기 반응 시간을 줄일 수 있는 Phased RGSS 방법을 제안하고, 그의 버퍼 요구량 및 초기 반응 시간에 대해 분석하였다.

2. 배경 연구

2.1 GSS(Group Sweeping Scheduling)

연속 저장 매체에 대한 전형적인 디스크 스케줄링 방법으로는 고정된 순서에 의한 방법과 SCAN의 두 가지가 있다. 고정된 순서에 의한 스케줄링 방법은 각 스트림에 할당된 버퍼에 데이터가 차가만 하면 전송이 가능해진다. 스트림이 n 개이고 하나의 버퍼 크기가 m 이라고 했을 때 고정된 순서에 의한 스케줄링 방법은 $(n+1)m$ 크기의 버퍼를 요구하게 된다. SCAN은 스트림의 재생 순서를 한 주기 동안 인출되어야 하는 물리적인 데이터 블록들의 위치에 의존하도록 하는 방법이다. 일반적으로 스트림에 할당된 블록들이 연속적이지 않기 때문에 스트림의 재생 순서는 매우 가변적이게 된다. 최악의 경우 이전의 주기에서 가장 먼저 데이터 블록을 읽은 스트림이 다음 주기에서 가장 나중에 읽을 수 있기 때문에 각 스트림에 대해 최고 두개의 버퍼 공간이 필요하게 된다. 관리의 편의를 위하여 SCAN은 각 주기가 끝나는 시점에서 모든 스트림의 재생이 시작된다. 이 방법은 $2nm$ 의 버퍼크기를 요구하게 된다. GSS[3]는 고정된 순서에 의한 스케줄링 방식과 SCAN방식을 공식화 한 것으로 버퍼 요구량을 줄일 수 있는 특성을 지니고 있다. GSS는 전체 스트림을 g 개의 그룹으로 나누어서 그룹들 사이에는 고정된 순서에 의한 스케줄링 방법을 사용하고 그룹 내에서는 SCAN에 의한 스케줄링 방법을 사용한다. 이 방법은 $(n + \lceil n/g \rceil)m$ 크기의 버퍼를 요구한다. 그림 1은 GSS에서 하나의 그룹이 세 개의 스트림으로 구성되었을 때 재생이 시작되는 시점과 버퍼가 해제되는 시점을 보여준다.

필요한 버퍼의 크기를 줄이기 위해서는 디스크 헤드의 탐색 시간(seek time)을 최소화하고, 한 스트림을 서비스하기 위해 버퍼에 데이터를 읽어들이는 작업이 되도록 일정한 간격(재생 주기)을 두고 일어나도록 해야 한다. n 개의 스트림이 있다고 가정하면 고정된 순서에 의한 스케줄링 방식은 한 스트림을 위한 재생 시지의 간격이 일정하다는 점에서는 버퍼

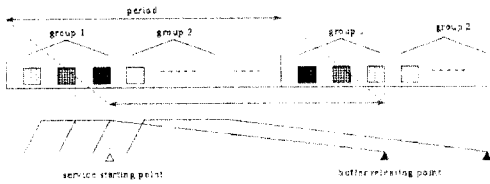


그림 1 GSS의 서비스와 버퍼 조작 시기

요구량에서 유리하다. 반면 디스크 상에 데이터의 위치가 임의적이기 때문에 탐색시간이 늘어나서 재생 주기가 길어지게 되므로 버퍼 요구량에서 불리하다. SCAN 방식은 그 반대의 현상이 발생한다. 즉, 탐색 시간은 최소가 되므로 버퍼량에 유리하게 작용하나 재생 간격은 최아의 경우 한 스트림을 위한 데이터 블록을 연달아 두 번 읽게 되므로 버퍼 요구량에서 불리해진다. GSS는 $g = n$ 인 경우 고정된 순서에 의한 접근 방법과 동일하게 되며, $g = 1$ 인 경우는 SCAN과 동일하다. 따라서, g 의 선택에 따라 탐색 시간과 재생 간격을 조절할 수 있고, 이에 따라 버퍼 요구량도 달라진다.

2.2 Region-based Data Layout (영역 기반 데이터 배치 방법)

서로 다른 사용자 스트림이 요구하는 멀티미디어 파일들은 서로의 저장위치에서 지역성을 가지지 않는다. 비록 동일한 사용자의 두 번 이상에 걸친 데이터의 요구에 대해서는 지역성을 가질 수는 있지만 시간 상으로는 서로 멀리 떨어져 있다. 이러한 특성을 이용하여 [8]은 한 디스크의 공간을 R 영역으로 나누고 스트림 그룹들이 각 재생 주기동안 접근할 데이터를 한 영역 안에 배치하는 영역 기반 데이터 배치 방법을 제안했다. 하지만 이 방법은 사용자가 처음으로 요구를 보낸 뒤 첫 플레이 백이 이루어지기까지의 시간으로 나타나는 초기 반응 시간이 길어진다는 단점이 있다.

그림 2는 디스크가 네 개이고 영역의 개수가 10인 경우에 대해 [7]이 제안한 데이터 배치 형태를 나타낸다.

2.3 RGSS (Region-based GSS)

GSS방식에서 재생을 위한 주기는 g 개의 스트림 그룹마다 필요한 디스크 전 범위의 탐색시간(full-stroke seek)으로 이루어져 있다. 즉, 스트림 그룹과 사용자 스트림이 증가할 수록 재생에 필요한 주기가 길어진다. 따라서 GSS방식은 스트림의 개수가 증가하면 g 가 1에 가깝게 되어 SCAN처럼 동작하게 되는 단점을 가지게 된다. 만약 각 스트림 그룹이 한 재생 주기 동안 디스크의 일정 영역만을 접근하도록 한다면 탐색시간에 의한 오버헤드를 크게 줄일 수 있게 된다.

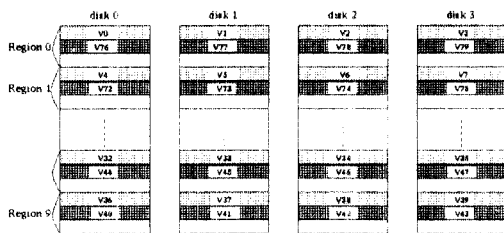


그림 2 영역 기반 데이터 배치

RGSS[7]은 [8]이 제안한 영역에 기반한 새로운 데이터 배치방식과 [3]이 제안한 GSS 방식을 접목시킨 것으로 stream의 수가 증가할 때 SCAN처럼 수행되는 GSS에서의 단점을 제거할 수 있는 것으로 분석되었다. 하지만 영역기반 데이터 배치방식이 가지는 초기 반응 시간에 대한 문제점은 여전히 가지게 된다.

그림 3은 GSS와 RGSS가 데이터를 읽어들이기 위한 디스크에 접근 방법을 보여준다. 각각 5개의 스트림 그룹에 대해 GSS는 한 주기동안의 데이터 검색을 나타내고, RGSS는 2x10 주기 동안의 데이터 검색을 나타낸다. 그림에서 GSS는 디스크의 전 범위에 대해 스트림 그룹의 개수만큼 SCAN이 이루어지지만, RGSS의 경우 영역의 범위에서만 SCAN이 이루어지므로 탐색시간에 의한 오버헤드가 줄어드는 만큼 버퍼의 요구량이 줄어들게 된다.

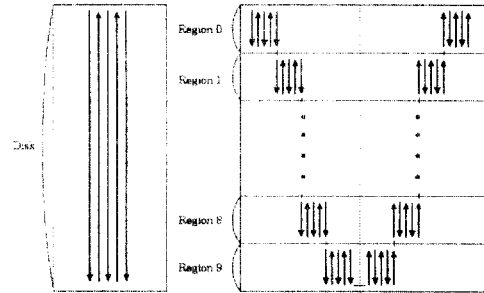


그림 3 GSS vs RGSS : 데이터 검색

3. Phased RGSS

RGSS 방식은 한 주기에 하나의 영역만을 검색하게 된다. 따라서 전체 R개의 영역이 있을 때, 임의의 영역을 검색하기 위해서는 최대 2R개의 다른 영역을 방문한 뒤야 가능하게 된다(그림 3). 이것은 한 영역에서 읽어들이는 데이터를 재생하는데 걸리는 시간(한 프레임은 서비스하는 시간)을 T_{frame} 이라고 하면 사용자로부터 새로운 스트림의 요구가 들어왔을 때 R개의 전체 영역에 대해서 최대 $2RT_{frame}$ 시간만큼, 평균적으로는 RT_{frame} 시간만큼 기다린 후에야 비로소 요구된 새로운 스트림을 시작할 수 있다는 것을 의미한다. 이러한 제한은 실제 VOD(Video On Demand)를 운영하는 데 있어 크게 영향을 미칠 수 있는 요소이다. 비디오를 선택한 후 몇 분이 지나도록 영화가 시작되지 않는다면 사용자는 짜증이 나거나 자신이 무엇인가 잘못하지 않았는가 하는 생각을 가지게 될 것이다.

본 논문에서는 RGSS에서 초기 반응 시간을 줄일 수 있는 새로운 방법을 제안한다. 앞에서 설명한 것처럼 RGSS는 읽어들이는 데이터를 재생하는 시간(T_{frame})과 한 영역에서 모든 스트림의 데이터를 읽는데 필요한 시간을 끝까지도둑 하고 있다. Phased RGSS의 동작이 되는 것은 이 두 가지의 시간을 서로 다르게 하였을 때, 특히 T_{frame} 을 RGSS와 동일하게 하고 한 영역에서 데이터를 읽어들이는데 필요한 시간을 줄였을 때, 새로운 스트림의 요구에 대한 초기 반응 시간을 줄여보고자 하는 것이다.

그림 4는 RGSS와 Phased RGSS의 차이를 살펴보기 위한 간단한 예이다. 네 개의 영역으로 나누어진 디스크에 대해 RGSS와 Phased RGSS 각각은 3개의 스트림 그룹을 가진다고 하자. 화살표는 한 영역에 대해서 세 개의 스트림 그룹이 데이터를 SCAN하는 방향을 나타낸다. 그림은 세 개의 스트림 그룹들이 RGSS에서는 같은 영역 내에서, 그리고 Phased RGSS에서는 다른 영역에서 데이터를 읽게 됨을 보여준다. 하지만 디스크 탐의 이동이 RGSS와 Phased RGSS 모두 동일한 거리만큼 연속적으로

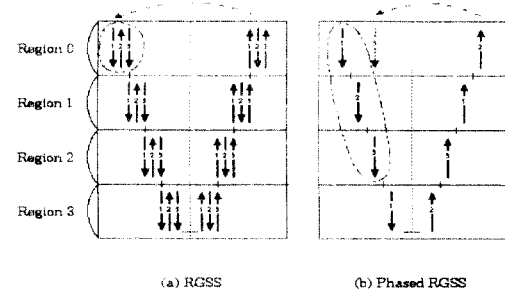


그림 4 RGSS vs Phased RGSS : 데이터 검색

이루어짐을 알 수 있다. 또한 디스크에서 데이터를 읽어들이는 스트림 그룹의 순서도 동일하게 나타나고 있음을 알 수 있다.(1,2,3,1,2,3,1,...) 이

것은 Phased RGSS가 RGSS와 동일한 버퍼량을 요구하면서 RGSS의 동일한 동작을 영역상의 위상차를 가지고 수행하고 있음을 의미한다.

일관성의 잃어버림 없이 모든 스트림의 처음 부분이 첫 번째 영역에 있다고 가정할 수 있다. 그러면 새로운 스트림의 요구가 들어왔을 때 첫 번째 영역에 접근할 때까지 이 새로운 스트림에 대한 서비스는 보류된다. 그림 4에서 RGSS의 경우에 비해 Phased RGSS의 경우에는 디스크 암이 다시 첫 번째 영역으로 돌아오는 데는 1/3의 시간밖에 걸리지 않는다. 따라서 새로운 스트림이 들어왔을 때 Phased RGSS의 방법이 RGSS 보다 빨리 서비스 할 수 있게 되는 것이다.

그림 4를 자세히 살펴보면 데이터 프레임의 위치가 RGSS와 Phased RGSS에서 서로 다를 수 있다. 즉, 1번 스트림 그룹이 데이터를 읽어들이는 영역의 순서가 RGSS와 Phased RGSS는 서로 다르다. 이것은 Phased RGSS에서 한 영역에서 데이터를 읽어들이는데 필요한 시간을 T_{frame} 보다 작게 하면서도 RGSS와 같은 순서의 재생이 가능하도록 데이터 배치를 수정하였기 때문에 발생한다. 그림 5는 그림 4의 구조에 대한 데이터 배치의 예이다. 이해의 편의를 위해 디스크의 암이 내려가는 방향에 대해서는 왼쪽에, 올라가는 방향에 대해서는 오른쪽에 표시하였다. 화살표는 RGSS와 Phased RGSS의 한 스트림에 대해서 어떻게 연속적으로 스트림의 재생이 일어나는지를 보여 준다. 여기서도 Phased RGSS의 디스크 암이 다시 첫 번째 영역으로 돌아오는데 걸리는 시간이 RGSS의 1/3 밖에 걸리지 않는 것을 볼 수 있다.

일반적인 Phased RGSS의 경우에 데이터를 배치하는 방법은 일정한 순서를 가지면서 모든 영역에 고루 데이터를 분산할 수 있는 것이어야 한

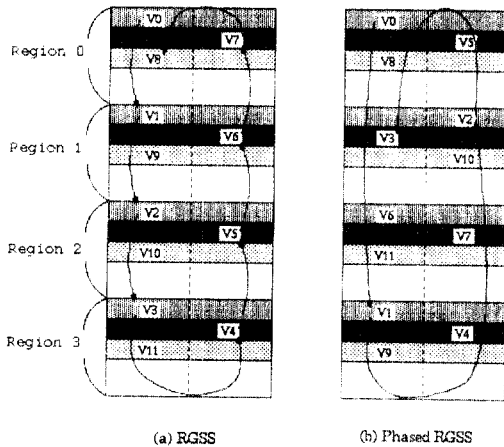


그림 5 RGSS vs Phased RGSS : 데이터 배치

다. 제한하는 방법은 데이터 배치 상수(c)라는 것을 두고 그 숫자의 정수 배를 따라 데이터를 배치하는 것이다. 데이터 배치 상수는 3이상의 R의 배수가 아닌 홀수라면 가능하다. 그림 5의 Phased RGSS의 그림은 데이터 배치 상수가 3인 경우에 대하여 데이터를 배치한 것과 같다.

4. 성능 분석

4.1 버퍼 요구량 분석

여기서는 스트림의 정상적인 재생을 유지하기 위해 얼마만큼의 데이터를 읽어서 버퍼에 유지시켜야 하는지를 살펴본다. 그림 6은 GSS 등의 방법에 대한 버퍼 요구량 분석을 위한 그림이다. Group i에 대한 비디오 세그먼트들은 t_{i+1} 의 위치에서 채워지고 이때부터 재생되기 시작해서 다음 주기의 t_{i+1} 에서 버퍼가 비워진다. 최악의 경우 다음에 재생될 그룹의 스트림의 내용을 디스크를 통해 버퍼로 읽어들이는 동안 서비스 가능한 모든 스트림에 대해 버퍼가 활용되어 재생되고 있어야 한다. 따라서 GSS, RGSS, Phased RGSS 모두 $n + s (= (g + 1)s)$ 개의 버퍼 블록이 필요하게 된다.

버퍼 블록의 크기(m)는 스트림이 재생되는 시간(mT_s)과 디스크에서 스트림을 읽어들이는데 필요한 시간에 의해 결정된다. 일반적으로 최악의 경우에도 정상적인 서비스를 보장하기 위해서는 스트림이 서비스되

는 시간이 모든 스트림을 읽어들이는데 필요한 시간보다 크거나 같도록 설정하여야 한다.

GSS, RGSS, Phased RGSS 모두 기본적으로 각 스트림에 대해 m 만큼 읽어들이는 시간이 필요하다. 이것은 m이 차지하는 디스크 트랙의 수와 한 트랙을 읽어들이는데 필요한 시간, 그리고 전체 스트림의 수의 곱으로 표현된다. 순수한 디스크 읽기 작업 외에도 그룹간의 SCAN에 드는 오버헤드가 필요하다. 모든 스트림을 읽어들이는데는 g 번의 디스크 SCAN이 필요하다. RGSS와 Phased RGSS와 같이 영역에 기반한 방법은 디스크 전체크기에 대해 $1/R$ 의 경의만큼만 SCAN 하면 되므로 GSS의 경우에서처럼 디스크 전체 영역을 SCAN해야 하는 것에 비해 최대 탐색 시간(seek time) S_{max} 가 $1/R$ 의 비율로 작아지게 된다. 그리고 마지막으로 디스크 암 스케줄러와 관계없이 요구되는 최소 탐색 시간이 필요하게 되는데 이것을 S_{min} 이라고 하면 GSS에서 서비스 가능한 전체 스트림들을 읽어들이는데 필요한 시간은 다음과 같다.

$$T_{RD_n} = n \left[\frac{m_{GSS}}{M_t} | t_r + g S_{max} + n S_{min} \right] \quad (1)$$

반면에 RGSS와 Phased RGSS에서는 최대 스캔시간 S_{max} 가 $1/R$ 의 비율로 작아지므로 전체 스트림들을 읽어들이는데 필요한 시간은 다음과 같다.

$$T_{RD_m} = n \left[\frac{m_{RGSS}}{M_t} | t_r + g \frac{S_{max}}{R} + n S_{min} \right] \quad (2)$$

$$T_{RD_{ph}} = n \left[\frac{m_{Ph}}{M_t} | t_r + g \frac{S_{max}}{R} + n S_{min} \right]$$

여기서 M_t 는 트랙당 저장 공간의 크기, t_r 은 디스크의 1회전 시간을 나타낸다. 앞에서 주지했던 것처럼 데이터의 연속적인 재생을 보장하기 위해서는 한 주기동안 스트림이 서비스되는 시간이 모든 스트림을 읽어들이는데 필요한 시간보다 크거나 같아야 할 필요가 있다. 즉 $mT_s \geq T_{RD}$ 이 유지되어야 한다. 그러면 식 1과 식 2로부터 RGSS와 Phased RGSS의 m을 다음과 같이 유도해 낼 수 있다

$$m_{GSS} \geq \frac{M_t}{M_t T_s} (g S_{max} + n S_{min})$$

$$m_{Ph} = m_{RGSS} \geq \frac{M_t}{M_t T_s - n t_r} (g \frac{S_{max}}{R} + n S_{min})$$

4.2 초기 반응 시간 분석

새로운 스트림이 들어왔을 때 평균적으로 얼마의 시간 동안 기다려야 서비스되기 시작하는지 살펴볼 것이다. RGSS의 경우 각 영역에서 각각의 스트림당 한 프레임에 해당되는 데이터를 디스크에서 읽어들이는데, 이때 필요한 시간은 한 스트림의 한 프레임이 재생되는 시간과 같은 길이를 가지게 된다. 이것을 그림 6에서 T_{frame} 으로 표시하였다. 따라서 새로운 스트림의 요구에 대해서 평균적으로

$$T = \frac{1}{2R} \left(\sum_{i=1}^{2R} i \right) T_{frame} = \left(R + \frac{1}{2} \right) T_{frame}$$

의 시간이 지나면 요구된 스트림이 처음으로 재생될 수 있다. Phased RGSS 방법에서 전체 영역을 R로 나누었을 때 디스크 헤드가 내려가는 방향과 올라가는 방향을 고려하면 전체 2R개의 논리적인 영역을 생각할 수 있다. 서비스를 위해 읽고 있는 영역(디스크 헤드가 위치하는 영역)이 전체 2R의 영역에서 i번째 영역이라 하고 이 때 새로운 스트림의 요구가 들어왔다고 가정하자.

스트림이 서비스되기 까지 기다려야 하는 시간은 새로운 스트림을 스트림 그룹에서 비어 있는 부분에 할당할 수 있을지 처음으로 검사할 때까지 필요한 시간과 첫 번째 영역에서 비어 있는 부분이 있는 스트림 그룹이 처음으로 서비스를 받게 될 때까지 걸리는 시간의 두 부분으로 나눌 수 있다. 편의상 전자를 T_1 으로 후자를 T_2 6666로 두겠다. T_1 은 현재의 위치에 있는 디스크의 헤드가 첫 번째의 영역을 읽을 수 있게 될 때까지 필요한 시간을 의미하므로, 현재의 위치가 전체 2R개의 논리적인 영역에서 i번째라 하면 $2R - i$ 개의 영역에 대한 서비스 시간이 된다. 현재의 위치가 2R의 영역에서 i번째에 있을 확률은 모든 i에 대해서 동일하게 $1/2R$ 이다. Phased RGSS에서 한 프레임이 서비스되는 시간(T_{frame})은 RGSS에서처럼 한 영역에서 소비하는 시간과 같은 것이 아니라 데이터 배치 상수 c개만큼의 영역에서 데이터를 읽어들이는데 필요한 시간과 같다. 따라서 한 영역에서 소비되는 시간은 T_{frame}/c 로 나타낼 수 있다.

새로운 서비스 요청이 들어 왔을 때 디스크가 첫 번째 영역으로 가서 비어 있는 스트림 공간이 있는지 검사하는데 까지 필요한 시간 T_1 은

$$T_1 = \frac{1}{2R} \sum_{i=1}^{2R} \frac{T_{frame} i}{c}$$

$$= \left(R + \frac{1}{2}\right) \frac{T_{frame}}{c}$$

이 된다.

만약 첫 번째 검사에서 비어 있는 스트림 공간이 있다면 새로운 서비스 요청에 대해 서비스하기까지 필요한 시간은 T_1 이면 충분하다. 하지만 스트림 그룹이 스트림으로 가득 차 있을 경우 다음 첫 번째 영역에서의 검사 때까지 기다려야 한다. 시스템이 최대 N 개의 스트림을 서비스할 수 있고, 현재 서비스되는 스트림의 개수를 n 이라 하면, $p (= n/N)$ 의 비율로 스트림 공간이 채워져 있다고 말할 수 있다. 한 영역에서 서비스되는 스트림 그룹의 묶음을 페이즈(phase) 그룹이라 하면 데이터 배열 상수가 c 일 때 전체 $\lceil N/c \rceil$ 개의 페이즈 그룹이 존재한다. 각 페이즈 그룹에는 $\lceil N/c \rceil$ 개의 스트림이 들어 갈 수 있으며 이를 k 라 하자. i 번째 페이즈 그룹에 이르러 비어 있는 스트림 공간이 있을 확률은 $i-1$ 번째 페이즈 그룹까지 모두 스트림이 가득 차 있을 확률(P_a)과 i 번째 페이즈 그룹에 빈 스트림 공간이 있을 확률(P_b)의 곱이다. P_a 와 P_b 는 둘 다 hyper geometric distribution을 따르며,

$$P_a(i) = \frac{\binom{n}{(i-1)k}}{\binom{N}{(i-1)k}}$$

$$P_b(i) = 1 - \frac{\binom{n-(i-1)k}{k}}{\binom{N-(i-1)k}{k}}$$

로 주어진다. 그러면 처음으로 비어있는 스트림 공간이 나올 때까지 필요한 평균 시간 T_2 는

$$T_2 = \frac{1}{c} \sum_{i=1}^{\infty} 2R \frac{T_{frame}}{c} i P_a(i) P_b(i)$$

$$= \frac{2RT_{frame}}{c^2} \sum_{i=1}^{\infty} i P_a(i) P_b(i)$$

가 된다. 따라서 p 의 확률로 스트림들이 채워져 있을 때, 새로운 스트림의 요구는 평균적으로

$$T = T_1 + T_2 = \left(R + \frac{1}{2}\right) \frac{T_{frame}}{c} + \frac{2RT_{frame}}{c^2} \sum_{i=1}^{\infty} i P_a(i) P_b(i)$$

의 시간이 지난 후에 서비스된다.

그림 7은 서비스 가능한 최대 스트림의 수가 50개이고 디스크를 8개의 영역으로 나누었을 때, 데이터 배치 상수에 따른 평균 대기 시간을 계산한 것이다. 제안된 Phased RGSS 방법은 RGSS와는 달리 현재 서비스되는 스트림의 수에 따라 기다려야 하는 시간이 달라진다. 또한 데이터 배치 상수에 따라라도 기다려야 하는 시간이 크게 차이가 남을 알 수 있다. 데이터 배치 상수가 충분히 큰 경우에는 현재 서비스되고 있는 스트림의 개수에 상관없이 기다려야 하는 시간이 RGSS보다 항상 작게 됨을 볼 수 있다. 이것은 영역에 의한 데이터 배치 방식으로 GSS를 설계할 경우, 데이터 배치 상수의 크기를 조절함으로써 항상 RGSS보다 높은 성능을 유도할 수 있음을 의미한다.

5. 결론 및 향후 연구

본 논문에서는 연속매체 인출에 있어 좋은 성능을 보이는 GSS 방법을 개선시킨 RGSS 방법에서, 새로운 스트림이 도착할 때 초기 반응 시간이 커지는 문제점을 해결하였다. GSS에 대해 버퍼 요구량을 줄인 RGSS의 장점을 유지시키면서 새로운 스트림에 대한 초기 반응 시간을 크게 감소시키는 Phased RGSS 방법을 제안하였으며, RGSS 방법과의 성능 비교를 위해 요구 버퍼량과 초기 반응 시간에 대한 분석을 하였다.

제안된 방법은 데이터를 각 영역에 분산시키면서 한 영역에서 데이터를 읽어 들어오는 시간을 결정하게 되는 데이터 배치 상수에 크게 의존하게 되는 특성을 지닌다. 이 상수를 충분히 크게 해 줄으로써 기존의 RGSS 보다 항상 초기 반응 시간이 작아지도록 할 수 있음을 보여 주고 있다.

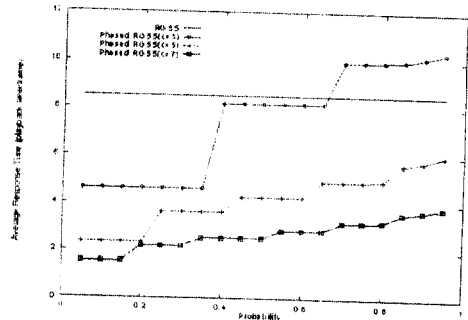


그림 6 $N=50$, $R=8$ 인 경우에서 데이터 배치 상수에 따른 Phased RGSS 방법의 초기 반응시간

향후 제안된 Phased RGSS 스케줄링 방법이 가정하는 동정의 스트림들을 확장하여 이걸의 스트림들에도 적용되는 확장된 방법에 대한 연구가 필요하며, 또한 버퍼 개성 정확과의 연관성에 대한 연구도 필요하다.

참고 문헌

- [1] A. L. N. Reddy and J. Wyllie, "Disk scheduling in a multimedia i/o system," Intl. Conf. on Multimedia, Aug. 1993.
- [2] D. J. Gemmell and S. Christodoulakis, "Principles of delay-sensitive multimedia data storage and retrieval," ACM Trans. on Information Systems, vol. 10, Jan. 1993.
- [3] P. S. Yu, M. S. Chen, and D. D. Kandlur, "Grouped Sweeping Scheduling (GSS) for DASH Based Multimedia Storage Management," tech. rep., IBM T. J. Watson Research Center, 1994.
- [4] F. A. Tobagi, J. Pang, R. Baird, and M. Gang, "Streaming RAIDTM: a disk array management system for video files," Proceedings of the 1st Intl. Conf. on Multimedia, Aug. 1993.
- [5] P. Y. Rangan, H. M. Vin, and S. Ramanathan, "Designing an on-demand multimedia service," IEEE Communication Magazine, July 1992.
- [6] D. D. Kandlur, M. S. Chen, and Z. Y. Shae, "Design of a Multimedia Storage Server," tech. rep., IBM T. J. Watson Research Center, 1993.
- [7] D. Y. Kang and C. I. Park, "An Efficient Disk Array Management for Video Server to Minimize Buffer Requirement," tech. rep., Department of Computer Science/PURL POSTECH, 1995.
- [8] Y. J. Oyang, M. H. Lee, and C. H. Wen, "A Video Storage System for On-Demand Playback," tech. rep., National Taiwan University, 1994.