

디스크 파티션내 고정 블록에 기반한 다중 스트림의 효율적 저장 방식

正會員 최 성 욱*, 박 승 규**, 최 덕 규**

An Efficient Storing Method of Multiple Streams based on Fixed Blocks in Disk Partitions

Sung-Wook Choi*, Seung-Kyu Park**, Dug-Kyoo Choi** *Regular Members*

※본 논문은 과학재단의 특성기초연구(96-01-01-05-1-3)에 의해 지원 받았음.

요 약

최근 급격히 발전된 컴퓨터 기술에 힘입어 다양한 분야에서의 멀티미디어 데이터 처리가 가능하게 되었다. 그러나, 기존의 디스크 저장 시스템으로 멀티미디어의 특징을 수용하기에는 무리가 있다. 현재, 몇몇의 연구에서 멀티미디어 저장에 관한 향상된 기술을 소개하였고, 그중 Bocheck는 액세스 주기와 단위가 동일한 다중스트림의 분할 저장 기법을 제안하였으나, 서로 상이한 주기를 갖는 연속 미디어의 스트림에 관해서는 고려되지 않았다. 이 논문에서는 조화 주기가 서로 상이한 스트림을 고정된 몇 개의 블록으로 할당하는 방안을 제안하고 주어진 다중 스트림을 m개의 시퀀스로 스케줄링시 그 가능성을 분석하였으며, 기존의 Scan-EDF와 디스크 분할 저장 방식과 비교하기 위해서 시뮬레이션을 행하였다.

ABSTRACT

Recent evolution in compute technology makes the multimedia processing widely available. Conventional storage systems do not meet the requirements of multimedia data. Several approaches were suggested to improve disk storing methods for them. Bocheck proposed a disk partitioning technique for multiple streams assuming that all streams have same retrieval intervals with the same amount data for each access. While Bocheck's one provides a good method for same period, it does not consider the case of different periods of continous media streams. This

*시립인천전문대 전산과

**아주대학교 정보 및 컴퓨터공학부 교수

論文番號:96364-1118

接受日字:1996年 1月 18日

paper proposes a new partitioning technique in which a fixed number of blocks are assigned for the streams with different retrieval periodicity. The analysis shows this problem is the same as the one scheduling the streams into a given sequence. The simulation was done to compare the proposed m-sequence merge method with the conventional Scan-EDF and Partitioning methods.

I. 서 론

최근의 컴퓨터 시스템의 발전으로 인하여 다양한 폭넓은 멀티미디어의 응용이 가능하게 되었다. 이 분야에서 중요한 관심 중의 하나는 연속된 다중스트림 즉, 비디오와 오디오 데이터 등 실시간 재생 요구에 부합되는 효과적인 디스크 내의 자료 저장 방법이라고 할 수 있다. 종래의 화일시스템 처리에서 사용되고 있는 방법인 회전하는 디스크의 표면을 헤드가 임의적으로 움직이면서 요구된 데이터를 검색하는 방식으로는 현재의 멀티미디어 데이터의 광범위한 특성을 수용할 수 없다 [11, 12].

이를 위한 멀티미디어 파일 기록 구조에 대한 새로운 방안들이 제안되었는데 대표적인 방식을 소개하면, 마감시간우선(EDF) 방식을 확장하여 고안된 마감 시간 우선 주사(SCAN-EDF) 방식[13]은, 고전적인 디스크 스케줄링 방법인 스캔과 실시간 보증을 위한 마감 시간 우선 방식을 조화한 방법으로 동일한 유사 마감 시간이 많을 때 접근 시간이 빠른 반면 시스템 환경 능력에 따라 스캔시간의 부가로 인한 처리 능력에 차이가 난다.

집단 일소(Group Sweeping) 방식 [14]은 디스크 압의 동작을 최소화하기 위하여 각 스트림을 N개의 고정된 순서로된 몇 개의 집단으로 분류하여 처리하는 방식으로 순환 시간(Cycle-Time)과 스트림의 자료 율에 따라 적절히 대응하는 버퍼가 있어 연속 처리를 가능하게 하나, 스트림을 동기화 할 경우 관련 스트림의 근접 배치에 어려운 점이 있다. 그 외 최근에 제안된 디스크 배치 방법으로는 대표적으로, Bocheck이 제안한 디스크 분할(Disk Partition) [4] 기법을 들 수 있는데, 미디어들이 같은 주기 및 같은 추출 단위를 갖는 것으로 가정하고, 디스크를 몇 개로 분할하여 종방향 우선으로 주사시키는 방식이며, 횡(Track)방향은 다수의 다른 스트림을 배치할 수 있기 때문에 상대적으로 적은 용량의 버퍼가 사용된다. 이 방식은

복수 스트림의 동시 전송을 필요로 하는 멀티미디어 서버용 디스크에 적합하지만, 주기가 서로 다른 동기화 스트림의 배치에는 적절하지 않다. 또한 이러한 제한을 없애고, 다양한 주기 및 다양한 추출 단위에 적합한 디스크 분할 기법이 제안된 바 있으나 [17] 분할된 스트림에 대응되는 버퍼의 크기가 일정하지 못하고 비효율적이다.

본 논문에서는 멀티미디어 서버에 알맞는 디스크 저장 방식인 디스크 분할 기법을 개선하고, 주기가 서로 다른 복수의 연속 미디어(CM:Continuous Media) 처리에 장점을 가지며, 또한 고정된 크기의 효율적인 버퍼로 주기화된 스트림 상호간의 동기화를 신속히 할 수 있는 새로운 디스크 저장 방식을 제안한다. 이를 위하여 본 논문에서는, 2장, 3장에서는 디스크 분할 기법의 개념과 동기성 다중스트림의 디스크 분할 배치의 일반적인 방법과 그 문제점을 분석하고, 4장에서 멀티미디어 서버를 위한 개선된 디스크 저장 방식을 제안한다. 5장에서는 시뮬레이션 및 분석을 하고, 6장에서는 간략히 본 논문의 결론을 내린다.

II. 디스크 분할 저장 기법

연속 미디어의 단일 형태의 스트림을 위한 저장 방식은 단일 디스크 표면에 n개의 원형 부분들 (1, 2, 3, ..., n)인 파티션으로 나누고 각 파티션은 여러 개의 트랙들로 구성되며, 각 트랙은 고정된 블록으로 나누어진다. 그림 1은 4개 파티션의 경우에 대한 단일 미디어의 저장 방식을 보여주고 있다 [4].

각 스트림은 일련의 연속된 블록으로 표현되며, 스트림의 연속된 블록은 각각 다른 파티션에 기록되기 때문에 헤드가 특정 파티션 내의 스트림의 시작 지점을 만나면 동일 방향으로 (디스크 안쪽에서 바깥쪽으로 혹은 그 반대 방향으로) 디스크 표면을 읽어 간다. 주사 방법은 왕복 주사(Scan) 방법이 적용되며 본 논문에서는 편의상 동일 주사 방향에 해당되는 각 파티

선별 특정블럭의 집합을 시퀀스라 명명하였다. 디스크 분할 저장 방식의 예가 그림 2에 있다.

헤드는 항상 특정 방향으로 마지막 파티션의 끝에도달할 때까지 디스크를 읽어 가기 때문에 하나의 스트리pm을 읽는 동안에 다른 스트리pm의 검색 요청이 발생하면, 요청된 스트리pm의 시작 블럭이 있는 곳까지 현재의 스트리pm을 계속 읽으며, 헤드를 진행시키다가, 요청된 새로운 스트리pm을 만나면 그 지점에서 새로이 요청된 스트리pm까지 함께 읽는다. 이 방식은 새롭게 요청된 스트리pm의 시작 블럭을 찾을 때까지, 일반적인 블럭의 검색 시간인 최대 회전 시간과 최소 접근 시간의 합을 현재 헤드가 위치한 파티션과 요청된 블럭이 기억된 파티션까지의 파티션 개수를 곱한 값만큼의 지연 시간이 발생되지만, 일단 검색이 시작되면 병행 전송이 가능하기 때문에 멀티미디어 서버에서 매우 유리한 방식이 될 수 있다.

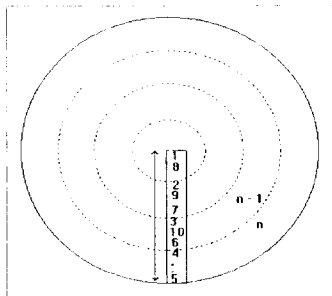


그림 1. 디스크 파티션

	블록0	블록1	블록2	블록3	블록4
파티션 0	$S_{1,1}$	$S_{1,K}$			$S_{2,10}$
파티션 1	$S_{1,2}$	$S_{1,K+1}$			$S_{2,9}$
파티션 2	$S_{1,3}$	$S_{1,K-2}$			$S_{2,1}$
파티션 3	$S_{1,4}$	$S_{1,K-3}$			$S_{2,2}$
파티션 4	$S_{1,5}$	$S_{1,K-4}$			$S_{2,3}$
파티션 5	$S_{1,6}$	$S_{1,K+5}$			$S_{2,4}$
	$S_{1,7}$	$S_{1,K+6}$			$S_{2,5}$

그림 2. 디스크 저장 기법

Ⅲ. 다중 미디어의 디스크 분할 배치

3.1 단순 배치

멀티미디어 데이터는 단일 미디어뿐만 아니라 영상, 음성, 텍스트등 복합미디어스트리pm을 동기 화하여 처리하는 경우가 많으며 디스크 분할 저장 방법을 동기성 복합스트리pm의 디스크 저장시 에도 적용할 수 있다.

단순 배치는 동기화 주기가 동일한 스트리pm을 같은 트랙 내에 저장하는 방법이며, 그림 3a에서 나타난 바와 같이 요구 주기가 동일한 미디어인 경우에 동기화 미디어 스트리pm s={영상, 음성, 이미지, 텍스트}는 각각 동일한 트랙 상에 존재함으로 헤드의 이동 없이 신속하게 동기화를 위한 데이터의 전송을 실행할 수 있다.

그러나, 멀티미디어의 동기화 스트리pm들의 주기는 각 미디어의 특성상 동일하지 않는 경우가 대부분이므로 주기가 상이한 스트리pm을 단순 배치하였을 경우, 스트리pm들의 동기 시점에서 동기화 참여 스트리pm 블럭들이 서로 다른 트랙 및 파티션에 기억되어 있는 경우가 많아 디스크 분할 기법의 특징을 살릴 수 없다.

	블록 0	블록 1	블록 2	블록 3
파티션 0	영상 1	음성 1	이미지1	텍스트1
	영상 8	음성 8	이미지8	텍스트 8
파티션 1	영상 2	음성 2	이미지2	텍스트2
	영상 7	음성 7	이미지7	텍스트7
파티션 2	영상 3	음성 3	이미지3	텍스트3
	영상 6	음성 6	이미지6	텍스트6
파티션 3	영상 4	음성 4	이미지4	텍스트4
	영상 5	음성 5	이미지5	텍스트5

그림 3a. 동일한 주기를 갖는 동기화 스트리pm 배치

3.2 LCM 배치

동기화에 참여할 스트리pm의 주기가 상이한 경우에는 단순 배치 방법이 적절하지 않으며, 각 스트리pm의 주기를 염두에 둔 배치를 해야 한다. 이러한 배치를 함으로써, 주기가 상이한 다중 멀티미디어 스트리pm을

동기화할때 대상 스트림이 동일 파티션, 동일 트랙 내에 연속으로 저장되므로, 추가로 헤드를 움직일 필요 없이 신속하게 읽어 들일 수 있다. 그림 3b는 멀티미디어 스트림 $s = \{\text{영상, 음성, 이미지, 텍스트}\}$ 가 상이한 주기 $T = \{1, 2, 3, 3\}$ 을 가지고 있을 때, 주기를 감안한 배치 형태를 보여주고 있으며, 그림 3c에서는 그 스트림을 디스크 분할 배치하였을 경우를 나타내었다. 그러나, 그림 3c에서 보는 바와 같이 4개의 시퀀스내에 단편화 현상이 심하고, 그 단편화 현상은 주기의 크기에 비례함을 볼 수 있다. 또한, 데이터를 디스크에서 읽어 들일 때, 시간 축에 따라 읽어 드리는 블록의 개수가 심한 차이를 보이며, LCM(Least Common Multiple) 위치에서 최대의 크기가 되기 때문에 사용되는 버퍼의 크기도 LCM시와 동일한 크기가 되어야 한다. 그러나, 버퍼를 최대로 사용하는 경우는 LCM시밖에 없음으로 버퍼의 낭비를 가져올 수 있다. 결국, LCM배치는 다중 스트림을 함께 읽는 문제에서 액세스 시간을 단축할 수 있는 반면, 그 주기들의 LCM에서 집중화 현상(여기서는 파티션2)이 초

래되고 [17] 그 결과 디스크의 이용률이 감소되며, 또한 필요한 버퍼의 크기가 증가하며 해당 시간에 재생해야 하는 동기화 메커니즘도 복잡하게 된다. 특히 디스크 이용률의 감소는, 대용량을 조건으로 하는 멀티미디어 자료 저장 방식에 심각한 문제를 초래할 수 있으므로 다중 사용자를 위한 동기화 다중 스트림형태에 대처할 수 있는 새로운 디스크 저장 방식이 필요하다.

IV. m-시퀀스 병합 배치

4.1 개요

m-시퀀스 병합 배치는 디스크 분할 배치 방법이 동기성 다중스트림을 처리할 때 발생하는 디스크의 단편화 현상을 방지하고, 디스크 재생시에 사용 버퍼의 이용률을 높이기 위하여 동기화 다중 미디어 블록들을 마감 시간 위배 없이, n개의 미디어스트림을 최소 m개의 시퀀스로 병합하는 방법이다. 이 경우 각 파티션은 m개의 고정블록으로 이루어진다 ($n > m$). 이 방법은 멀티미디어 데이터의 요구 주기가 단위 시간의 정수 배이고, 모든 미디어의 요구는 주기적이며, 각 미디어 데이터 요구량은 동일하다는 제약이 있지만, 이 문제는 디스크 메커니즘 자체가 블록이라는 일정한 양의 단위로 구성되고, 읽혀지며, 그 블록에 접근하는 행위 자체가 주기라 볼 수 있고, 모든 미디어 데이터를 연속 처리하기 위해서는 주기적으로 접근되어야 함으로, 위의 조건은 거의 무의미하다 할 수 있다.

m-시퀀스 병합 스케줄링은 한 시퀀스내에 가능한 많은 스트림 블록을 마감 시간에 맞추어 배치하는 것이 목적이므로 이는 멀티미디어 서버에서의 실시간 테스크 스케줄링 문제와 동일한 성격을 갖는다. 여기에서 마감시간의 의미는, 멀티미디어 데이터 연속처리시 임의의 시간(주기) 내에 해당 스트림 블록이 디스크로부터 읽혀져야 하며, 그최종시간을 마감시간이라 한다.

주기적 타스크의 실시간 스케줄링 문제에서, C_i 를 i 번째 타스크에 수행 시간, T_i 를 i 번째 타스크의 주기라 하면, T_i 가 C_i 의 정수배 일 때 다음 식과 같은 조건에서 모든 타스크들의 스케줄링이 가능함을 증명하였다[11].

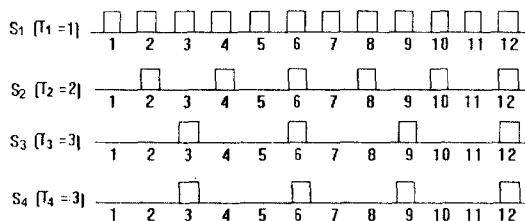


그림 3b. 동기화 다중 스트림의 LCM 배치

	블록 0	블록 1	블록 2	블록 3
파티션 0	영상 1			
	영상 8	음성 4		
파티션 1	영상 2	음성 1		
	영상 7			
파티션 2	영상 3		이미지1	텍스트1
	영상 6	음성 3	이미지2	텍스트2
파티션 3	영상 4	음성 2		
	영상 5			

그림 3c. 상이한 주기를 갖는 동기화 스트림 배치에서의 LCM 집중화

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_n}{T_n} \leq 1$$

그리고, 이 문제는 n개의 스트림을 한 개의 시퀀스에 병합할 수 있는가와 같은 문제가 된다. 모든 i에 대한 C_i 는 단위 블록과 같고 (즉, 모두 1이라 가정) 스트림의 주기 T_i 는 블록의 정수배 이므로 아래와 같은 조건이 성립하면 주기가 서로 다른 스트림을 한 시퀀스내에 병합할 수 있다.

스트림 S_1, S_2, \dots, S_n 의 주기를 T_1, T_2, \dots, T_n 이라 하면, 복합 스트림이 한 시퀀스내에 스케줄링될 필요

충분조건은 $\frac{1}{T_1} + \frac{1}{T_2} + \frac{1}{T_3} + \dots + \frac{1}{T_n} \leq 1$ 이다.

스케줄링 계수 E를 $E = \frac{1}{T_1} + \frac{1}{T_2} + \frac{1}{T_3} + \dots + \frac{1}{T_n}$

와 같이 정의하면, 시퀀스의 상태를 나타내는 스케줄링 계수는 다음과 같이 표현된다.

- $E < 1$ 인 경우 한 시퀀스내 단편화가 존재함으로 또 다른 스트림을 포함시킬 수 있다.
- $E = 1$ 인 경우 한 시퀀스내 스트림이 단편화 현상이 완전히 병합된 경우이며, 영상 등 짧은 주기를 갖는 단일스트림 이거나, 서로 다른 주기를 갖는 스트림들의 합이 이러한 값을 가지면 한 시퀀스내에 병합이 가능하다.
- $E > 1$ 인 경우 한 시퀀스내에서 복수 개의 스트림을 모두 배치할 수 없는 경우이며, 스트림을 분리시켜 별도의 시퀀스에 할당하여야 한다.

예를 들어, 그림 4a의 주기 T가 각각 3, 4, 4, 6인 동기화 복수스트림을 시퀀스 병합한 결과가 그림 4b, 그림 4c에 있다. 먼저, 동기화 복수스트림 S_1, S_2, S_3, S_4 의 스케줄링 가능성을 분석하여 보면, 스케줄링 계수는 $\frac{1}{3} + \frac{1}{4} + \frac{1}{4} + \frac{1}{6} = 1$ 으로 하나의 시퀀스내에 병합이 가능하다. 그림 4b, 그림 4c에서는 이러한 동기화 다중 스트림을 하나의 시퀀스에 병합한 결과를 나타내고 있다. 그러나, 그림 4b의 다섯 번째 블록 S_2 가 마감 시간을 위배하여 기어되어 있는 것을 볼 수 있다. 그것은, 스트림 S_2 의 주기는 4이며, 이는 다른 스트림 블록 (S_1, S_3, S_4)과의 동기화 작업을 위해서 첫

번째 단위시간부터 매4개의 단위시간 마다 최소한 1개는 디스크로부터 읽혀져야 하기 때문이다. 그러므로, m-시퀀스 병합 스케줄링시 스케줄링 계수가 $E \leq 1$ 일 경우라 할지라도 마감시간 위배를 방지하는 별도의 알고리즘이 필요하다.

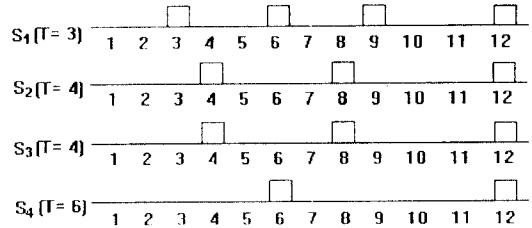


그림 4a. 4개의 스트림으로 구성된 S_1, S_2, S_3, S_4 의 동기화 배치

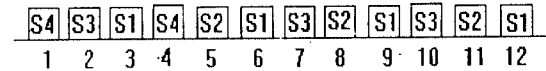


그림 4b. 1-시퀀스 합병된 S_1, S_2, S_3, S_4 (마감 시간 위배의 경우)

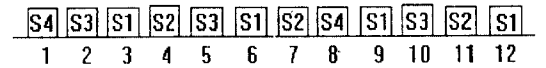


그림 4c. 1-시퀀스 합병된 S_1, S_2, S_3, S_4 (마감 시간 만족의 경우)

4.2 스트림 이동 배치

스트림의 이동 배치는 LCM 구간에서만 고려하기로 한다. 스케줄링 계수가 $E \leq 1$ 이고, 복수 개의 스트림이 마감 시간 스케줄링되어 있는 경우, 스트림의 집합을 $S = \{S_1, S_2, \dots, S_n\}$, 각각의 주기를 $T = \{T_1, T_2, \dots, T_n\}$ 이라 하고 $T_1 \leq T_2 \leq T_3 \dots T_{n-1} \leq T_n$ 일 때, 스트림의 최적 배치를 위한 방안으로써 다음과 같은 규칙을 설정한다.

(규칙1) 스트림을 마감 시간 배치하였을 경우 블록의 이동은 한쪽 방향으로 진행하며 그 이동 거리를 L이라 하였을 때, $L_i < T_i$ 이어야 한다.

(설명) 그림 4a와 같이 블록을 마감 시간 이후로 이동

하지만 최고 이동 거리가 T_i 보다 큰 경우 마감 시간 위배나 버퍼의 추가가 필요하다.

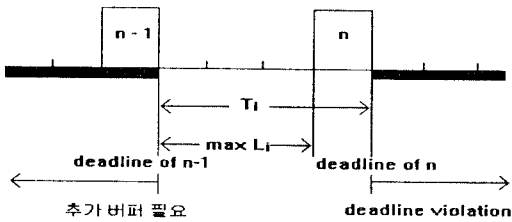


그림 4x. 마감 간 위배 현상

(규칙2) 복수 개의 스트림이 배치 중복이 발생하였을 경우, 이동 가능 길이가 짧은 스트림을 우선으로 배치 이동한다.

(설명) 주기가 각각 2, 3, 6인 3개의 스트림 S_1, S_2, S_3 를 하나의 시퀀스내에 배치할 경우에는 $LCM = 6$ 번째에서 3개의 블록이 중복된다. 여기서 주기가 2인 스트림 S_1 을 우선 배치하고 그 스트림내의 공백에 나머지 S_2, S_3 을 각각 병합하는 경우를 예로 들면

i) 긴 거리(L)를 우선으로 배치 이동 할 경우, 배치 이동 순서는 S_3, S_2 이 되며, 실제 이동 거리는 $L_3 = 1, L_2 = 3$ 이 된다. 그러나, S_2 의 주기는 3이므로 $T_2 = L_2$ 이 되고, (규칙1)을 위반한다(그림 4c).

ii) 짧은 거리를 우선으로 할 경우 배치 이동 순서는 S_2, S_2 이 되며, $L_2 = 1, L_3 = 3$ 이 되고, S_2 및 S_3 모두 (규칙1)를 만족한다(그림 4f).

결과적으로 그림 4g에서 보는 것처럼 배치 이동 스케줄링시에는, 중복된 다른 스트림의 블록을 하나씩 삽입시키므로, 우선 순위가 낮을수록 이동 길이 L이 길어지는 것은 당연하므로 $L_i < T_i$ 에 위배될 가능성이 크다.

순서 \ 주기	1	2	3	4	5	6
2		P_1		P_1	P'_3	P_1
6						P_3
3			P_2			P_2

그림 4e. 긴 이동 거리 우선 순위 배치

순서 \ 주기	1	2	3	4	5	6
2	P'_2	P_1	P'_3	P_1	P'_2	P_1
3			P_2			P_2
6						P_3

그림 4f. 짧은 이동 거리 우선 순위 배치

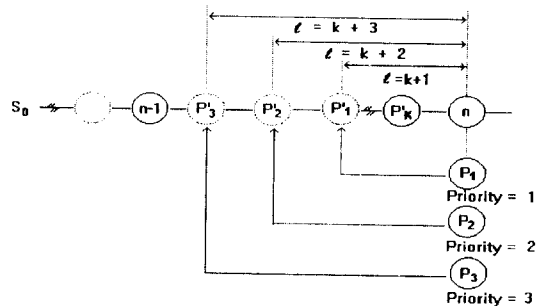


그림 4g. 우선 순위에 따른 배치 이동 거리 증가

4.3. m-시퀀스 병합 알고리즘

스트림 이동 배치 알고리즘은 다음 순서로 진행되며 그림 4h와 그림 4i에 실행 예를 나타내었다. 먼저 알고리즘에서 사용되는 변수의 내용은 다음과 같다.

k: 스트림 $S_1, S_2, S_3, \dots, S_n$ 주기의 LCM 그림 4h, 그림 4i에서 순위란의 12

p: 그림 4h, 그림 4i에서 스트림 블록이 기억되는 순서 (12에서 1까지 반복변환)

Dj: 스트림 $S_1, S_2, S_3, \dots, S_n$ 의 이동가능 길이이며 처음에는 해당 스트림의 주기가 초기치로 기억되고, 해당 스트림이 현재의 위치에 기억되지 못할 경우 1씩 감소한다.

그러나 규칙1) 에서 설명한 내용과 같이 각 스트림의 총 이동가능 길이는 스트림의 주기보다 클 수 없다. 그리고 해당 스트림블록이 p 번째에 할당되면, 진행방향에서 가장 가까운 다음번째 위치 p 에서 초기치인 주기가 다시 기억 된다.

또한 알고리즘의 수행은 그림 4h와 그림 4i에 나타

낸 것 과 같이 LCM 위치로부터 역으로 한단계씩 진행하면서 중복된 스트림 블록과 이전 단계에서 배치 받지 못한 스트림 블록을 포함하여 최소의 이동길이를 갖는 스트림블록을 현 위치에 배치한다.

```

let p = k;
k , be its LCM period;
for each stream Si
let the initial backward distance of the stream,
Di , be its period;
repeat
allocate to p-th partition with a block of stream
which has the smallest Di;
let p = p - 1;
for each unallocated stream Di
let Di = Di - 1;
for a stream which has p as its multiple of period, add it to a
unallocated stream set and let its backward distance be its period;
until p = 0;
    
```

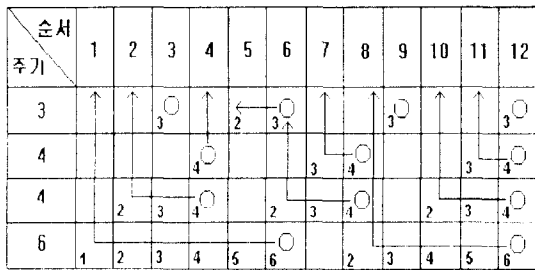


그림 4h. 주기가 3, 4, 4, 6인 다중 스트림의 이동 배치

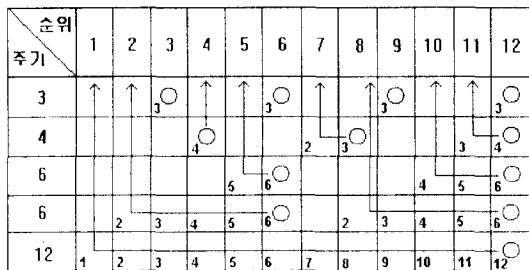


그림 4i. 주기가 3, 4, 6, 12인 다중 스트림의 이동 배치

V. 시뮬레이션 및 분석

본 논문에서 제안한 m-시퀀스 병합 저장 방식의 시뮬레이션을 위하여, 디스크 스케줄링의 대표적인 몇몇 기법의 검색 특성을 비교하였다.

본 논문의 주된 관심은 하나의 스트림으로 구성된

CM데이터의 서비스가 아닌, 복수개 동기화 스트림의 CM데이터서비스에 있으므로, 미디어의 연속 저장에 반드시 유리하지 않다.

마감 시간 우선 주사(SCAN-EDF) 방식이나, 집단 일소(GSS) 방식의 경우 동기화 스트림 작업 시에는 다음의 식과 같이 임의 블록주사 (RANDOM BLOCK SCAN) 방식과 같은 탐색 시간을 나타낸다.

$$T = B_n \{ S(B_i/T_r + R_x + SE) + 2E \} / 4$$

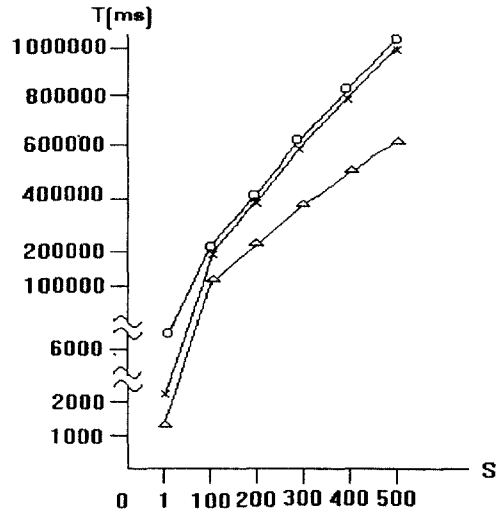
여기서 c 는 이동 실린더(cylinder)갯수 이고, SE 는 디스크 조사시간, a 는 헤드 안정 계수, b 는 가속 안정 계수이다. B_n 은 블록수, S 는 스트림수, B_i 은 블록 크기, T_r 은 최대전송율, R_x 은 디스크 최대 회전 속도, P 는 파티션 개수이다.

디스크 분할 기법은 $T = B_n \{ S(B_i/T_r + R_x + SE) + SE/P \}$ [4]로 표현되며 m-시퀀스 병합 배치 기법 역시 $T = B_n \{ S(B_i/T_r + R_x + SE) + SE/P \}$ 이지만 디스크 조사시간(SE)에서 마감 시간 우선 방식 및 집단 일소 방식은 단거리 탐색이나 장거리 탐색이며, 일반적인 디스크 분할 기법은 단거리 탐색이고, m-시퀀스 병합 배치 기법은 동기화 스트림이 동일 트랙 상에 존재함으로 조사시간이 거의 없거나, 최단거리 탐색으로 행할 수 있다. 또한 조사시간은 장거리 탐색일 경우 $SE = a + bc$, 단거리 탐색일 경우 $SE = a + b \sqrt{c}$ 이며, 최단거리 탐색일 경우 약 2ms가 소요된다.

시뮬레이션을 위한 파라미터의 설정은 디스크 시스템 HP97560을 기준 하였으며 주스트림의 예상 블록수는 각 100개, 파티션 개수는 16개로 하였고, 블록길이는 1028Byte로 정하였다. 그 외의 자세한 특징은 <표 1>에 나타내었다. 디스크 분할 저장시에 동기 스트림간의 근접 배치를 고려하지 않는 표준적 배치를 하였을 경우, 디스크 스케줄링 방식에 따른 검색 시간의 차이는 그림 5a와 같다. 결과에서 보듯이 와 같이 스트림수의 증가에 따른 디스크 검색 시간은 디스크 분할 저장 방식이 종래의 마감 시간 우선 주사 방식에 비하여 유리하게 나타나고, m-시퀀스 병합 방식은 각 동기화 스트림이 동일 트랙 내에 배치되고, 디스크 접근 시간이 최소 접근의 형태를 나타냄에 따라, 상대적으로 가장 좋은 결과를 보이고 있다.

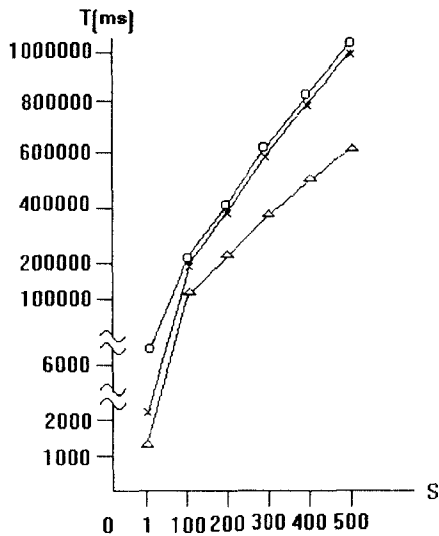
표 1. 파라미터 테이블

Parameter	Value
Sector size	512 Byte
Cylinders	1,962
Tracks per cylinder	19
Data sector per track	72
Number of zones	1
Track skew	8 sectors
Revolution speed	4,002 RPM
Controller reads	2.2 ms
Overhead writes	2.2 ms
sort (ms)	$3.24 + 0.400 \sqrt{c}$
Seek time long (ms)	$8.00 + 0.008c$
boundary	$c = 383$
maximum transfer rate	10M BPS



○ Scan-EDF × Partition △ m-Sequence Merge

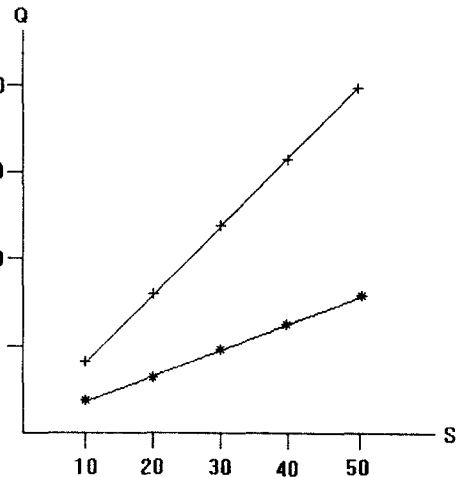
그림 5b. 디스크 검색 시간 비교 2



○ Scan-EDF × Partition △ m-Sequence Merge

그림 5a. 디스크 검색 시간 비교

디스크 분할 저장 방식에 동기화 배치를 하였을 경우에는 그림 5b와 같이 m-시퀀스 병합 저장 방식과 당연히 같은 검색 시간을 보이고 있지만, 이 경우에 디스크 사용율은 주어진 스트림 수 S에 대하여 예상되는 시퀀스 Q를 나타낸 것으로 그림 5c와 같이 디스크 분할 저장 방식에 비해 m-시퀀스 병합방식 좋은 결과를 나타내고 있다.



+ Partition * m-Sequence Merge

그림 5c. 디스크 이용률 비교

VI. 결 론

본 논문은 멀티미디어 서버에서 연속성을 갖는 동기성 스트림을 단일 시스템 형식으로 클라이언트에 전송하기 위하여 멀티미디어 서버에 존재하는 복수 개의 동기성 스트림 간에 동기화를 신속히 행하고 관

런된 시스템 자원의 효율성을 높이는 방안을 연구하였다. 연속성 데이터의 처리시 디스크 분할 방식의 특징은 임의로 요구되는 클라이언트들의 CM데이터의 요청을 디스크 내에서 헤드의 동작을 극소화시켜서의 동시에 읽어들일 수 있기 때문에 멀티미디어 시스템의 서버용 저장 방식에 유리하다. 그러나, 주기가 상이한 각각의 CM데이터들을 동기화 처리시 디스크 단편화 현상 및 출력 버퍼의 낭비가 발생한다. m-시퀀스 병합 방식은 디스크 분할 방식의 장점을 가지면서 동기성 CM데이터를 하나의 시퀀스내에 조합하여 디스크 및 버퍼의 사용을 높일 수 있다. 이를 위하여 본 논문에서는 복수 스트림을 단일 스트림의 형태로 병합하기 위한 병합 스케줄링 방안과 이 경우에 각 스트림의 마감 시간 위배를 방지하는 방안을 제시하고, 시뮬레이션 하였다. 먼저 기존의 디스크 스케줄링 방식과 비교할 때, 검색 시간의 경우 집단 일소 방식이나, 마감 시간 우선 주사 방식에 비하여 우수한 결과를 보였고, 디스크 분할 기법에 비하여 우수하거나 비슷한 결과를 보였는데, m-시퀀스 병합 저장 방식과 디스크 분할 기법이 비슷한 결과를 보인 경우는 디스크 분할 기법 적용 시에 디스크의 사용률을 무시한 채 동기화 배치에 중점을 둔 경우이며, m-시퀀스 병합 저장 방식이 우수한 결과를 보인 경우는 표준적인 디스크 분할 기법을 적용한 경우이다.

디스크 사용도 비교에서는 마감 시간 우선 주사 방식 및 집단 일소 방식은 일반적인 디스크의 사용을 전제로 하였기 때문에 비교 대상에서 제외 시켰고, 디스크 분할 기법에서의 동기화를 염두에둔 디스크 저장의 경우처럼 특수한 경우를 예로 하여 비교하였다.

결과적으로 m-시퀀스 병합 저장 방식은 멀티미디어 서버에서 복수 개의 스트림을 동기화 하여 다수의 사용자에게 전송할 경우에 검색 시간을 줄일 수 있으며, 디스크 사용 면에서도 유리하다.

향후 연구 과제로는 스트림 이동 배치 알고리즘의 공간 복잡도를 개선하는 문제와 본 논문에서 제안된 방식을 주문형 비디오(VOD) 시스템에 적용하는 방안 등이다.

참 고 문 헌

1. J. F. Allen, "Maintaining Knowledge about Tem-

poral Intervals", Communication of the ACM, pp. 832-843, 1983.

2. D. Anderson, "Meta-Scheduling for Continuous Media", University of California Berkeley.

3. D. P. Anderson, Y. Osawa, "A File System for Continuous Media", ACM Transactions on Computer Systems, Vol.10, No.4, pp.311-337, November 1992.

4. P. Bocheck, H. Meadows, S. Chang, "Disk Partition Technique for Reducing Meltimedia Access Delay", Preceedings of the IASTED/ISMM International Conference, pp.27-30, 1994.

5. J. I. Egan and T. J. Teixeira, Writing a UNIX Device Driver, John Wiley & Sons, Inc., 1988.

6. J. Gemmelle, S. Christodoulakis, "Principles of Delay-Sensitive Multimedia Data Storage and Retrieval", ACM Trans. on Information Systems, 51-90, 1992.

7. G. R. G. Bruce L. Worthington, R. Y. Hou, Y. N. Patt, "Disk Arrays", IEEE, Computer, pp.30-36, 1994.

8. H. I. Kim, H. S. Kim, C. S. Jhon, Y. K. Cho, "Considerations in Multimedia Storage System", Preceedings of International Workshop on INTEL-LIGENT COMPUTER, pp.110-124, Nov.1990.

9. P. Lougher, D. Shepherd and D. Pegler, "The Impact of Digital Audio and Video on High Speed Storage", Lancaster University, 1994.

10. P. V. Rangan, H. M. Vin, "Efficient Storage Techniques for Digital Continuous Multimedia", IEEE TRANS. ON KNOWLEDGE AND DATA ENGINEERING, pp.564-573, 1993.

11. A. L. N. Reddy, J. C. Wyllie, "I/O Issues in a Multimedia System," IEEE COMPUTER, pp. 69-74, March 1994.

12. C. Riemmer and J. Wilkes, "An Introduction to Disk Drive Modeling", IEEE Computer, Vol.27, No.3. pp.17-28, March 1994.

13. A. L. N. Reddy and J. wyllie, "Disk Scheduling in a Multimedia I/O System", Proc. 1st ACM Int'l Conf. on Mutimedia, ACM Press, New

York, pp.225-233, 1993.

14. M. -S. Chen, D.D. Kandlur, and P.S. Yu, "Optimization of the Group Sweeping Scheduling (GSS) with Heterogeneous Multimedia Streams". Proc. 1st ACM Int' I Conf. on Multimedia, ACM Press, New York, pp. 235-241, 1993.
15. 백윤철, 고진, "연속적인 미디어 검색을 위한 비 중단 실시간 태스크의 스케줄 가능성 분석", 정보 과 논문지 제21권 제7호, pp.1252-1260, 1994. 7.
16. 나인호, 박승규, 김성조, "사용자 개입 멀티미디어 프리젠테이션을 위한 시간 관계 명세 모델", 한국 정보과학회 학술 발표 논문집, pp.955-958, 1994년 도 춘계.
17. S.K.Park, J.W.Choi. "Storage Techniques for Continuous Media Streams with Disk Partitioning", JTC-CSCC'95, pp.447-450, Kumamoto, Japan, 1995.
18. Seung-Kyu Park, Sung-Wook Choi. "Storing Technique of Multiple streams on Disk with a fixed size Buffer for Playout", ITC-CSCC '96, PP.649-652, Korea, 1996.



최 성 욱(Sung-Wook Choi) 정회원
 1983년 광운대학교 전자계산과 이학사
 1987년 경희대학교 대학원 전자 공학과(전자계산전공) 공 학석사
 1997년 아주대학교 대학원 컴 퓨터공학과(멀티미디어

시스템전공) 박사과정수료

1989년~1992년 충남전문대학 전산과 조교수
 1992년~현재 시립인천전문대 전산과 부교수
 관심분야: 멀티미디어, 다중처리시스템, 객체지향시 스템



박 승 규(Seung-Kyu Park) 정회원
 1974년: 서울대학교 응용수학과 공학사
 1976년: 한국과학기술원 전산과 이학석사
 1982년: 프랑스 INPG ENSIMAG 전산학 박사
 1976년~1977년: 한국과학기술원 구소(KIST) 연구원
 1977년~1992년: 한국전자통신연구소(ETRI) 부장
 1984년~1985년: 미국 IBM 왓슨 연구소 연구원
 1992년~현재: 아주대학교 정보 및 컴퓨터공학부 교수
 관심분야: 멀티미디어 처리구조, 다중처리 및 MPP 시스템, 실시간 시스템, 심블릭 컴퓨터구 조, 이동컴퓨터시스템



최 덕 규(Dug-Kyoo Choi) 정회원
 1966년: 서울대학교 공과대학 원 자력공학(학사)
 1985년: 미국 Wright State Univ. 전산학(석사)
 1989년: 미국 Univ. of Massa- chusetts, Lowell 전산 학(박사)

1968년~1992년: KIST연구원, CDC Korea SE, 국방 과학연구소 책임연구원, 미국 Wright-Patterson AFB Avionics Lab 교환연 구원
 1992년 12월~현재: 아주대학교 부교수(고동기술연구 원 연구위원)
 관심분야: LAN, ATM, Mobile communications VOD