멀티미디어 서버의 저장 구조를 위한 디스크 배열의 성능 분석

조진성, 김태현^{o}, 김영구, 성민영, 신현식
서울대학교 컴퓨터 공학과

Performance Analysis of Disk Arrays for Storage Architecture of Multimedia Servers

Jinsung Cho, Taehyoun Kim, Youngku Kim, Minyoung Sung, Heonshik Shin
Department of Computer Engineering, Seoul National University

요 약

멀티미디어 서버는 취급하는 데이타의 크기가 매우 크며, 데이타의 실시간성을 만족시키기 위해 보장된 높은 입출력 대역폭을 필요로 한다. 이에 따라 멀티미디어 서버의 저장 구조로서 디스크 배열이 거론됨은 자명하다. 그러나 디스크 배열은 그 구성 방법 및 데이타의 할당 방법 등에 의해 성능의 차이를 보이므로 멀티미디어 응용과 같은 새로운 응용 분야에 적합한 모델에 대한 연구가 필요하다. 본 논문에서는 개인용 컴퓨터(PC)를 기반으로 한 멀티미디어 서버의 저장 장치로서 기존의 SCSI 어댑터와 SCSI 디스크로 구성되는 디스크 배열에 대한 성능을 분석한다. 성능에 영향을 미치는 요소로는 디스크 배열의 구성 방법, 디스크의 갯수, 디스크 스케쥴링, 스트라이핑 크기, 블럭 배치 방법 등을 들 수 있으며 이를 통해 멀티미디어 서버를 위한 최적의 저장 구조를 제시한다.

1 서 론

프로세서의 처리 속도에 비해 상대적으로 느린 디스크의 입출력 성능을 개선하기 위해 디스크 배열(disk array, RAID)이 등장하였다[1]. 현재 활발히 연구되고 있는 멀티미디어 응용 분야는 대용량의 데이타를 실시간으로 처리해야 한다. 따라서 대용량 고성능 디스크 장치를 필요로하는 멀티미디어 서버에서는 디스크 배열이 이용되고 있다. 그러나 디스크 배열의 성능을 완전히 발휘하기 위한 구성 방법 및 데이타의 할당 방법 등에 대한 연구는 부진한 실정이다.

[2]에서는 두개의 스트라이핑 디스크를 이용한 멀티미디어 서버의 구성에 대해 나와 있고, [3]에서는 멀티미디어 저장 시스템과 메모리 요구조건, 초기 지연시간(start-up latency)등을 분석하였다. [6]에는 디스크 배열에서의 멀티미디어 데이타 할당 방법에 대해 연구하였으나 멀티미디어 스트림의 연속성 요구조건을 간과한채 단순한 응답시간을 기준으로 하였다. [4]에서는 새로운 입출력 구조를 제안하고 비디오 서버의 각 요소에 대한 분석을 수행하였다.

본 논문에서는 PC를 기반으로한 멀티미디어 서버의 저장 장치로서 기존의 SCSI 어댑터와 SCSI 디스크로 구성되는 디스크 배열에 대한 성능을 분석하여 최적의 저장 구조를 제시한다.

본 논문의 구성은 다음과 같다. 2절에서는 고려하는 멀티미디어 서버 및 그 저장 구조에 대해 언급하고 3절에서는 이에 대한 실험 결과를 제시한다. 그리고 4절에서 결론을 맺는다.

2 고려된 멀티미디어 서버의 저장 구조

2.1 하드웨어 환경

본 논문에서 대상으로 하며 향후 구현할 멀티미디어 서버의 하드웨어 환경은 그림 1과 같다. 멀티미디어 서버는 PC를 기반으로 하며 프로세서, 메모리, 디스크 서브 시스템, 통신 서브 시스템, 시스템 버스 등으로 구성된다. CPU는 Pentium-100MHz이며 PCI 버스에 의해 메모리 및 입출력 서브 시스템과 연결된다. 디스크 서브 시스템은 SCSI-II 어댑터(adaptor)와 SCSI 디스크로 구성되며 디스크 배열의 제어는 CPU가 담당하는 소프트웨어적인 접근방법을 취한다. 그리고 멀티미디어 스트림의 메타 데이타는 디스크 서브 시스템의 대역폭을 보장하기 위해 로칼 디스크에 저장한다. 한편, 디스크 서브 시스템과 함께 멀티미디어 서버의 병목현상(bottleneck)은 통신 서브 시스템에서도 발생되나 본 논문은 고려하지 않기로 한다.

일반 응용과는 달리 멀티미디어 서버에서의 CPU 사용율 (utilization)은 그리 높지 않다. 비록 CPU가 디스크 배열을 제어한다고 하더라도 CPU는 모든 사용자의 요구를 각 디스크로 분배하

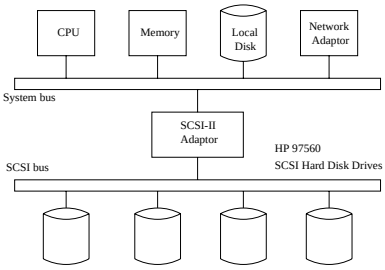


그림 1. 멀티미디어 서버의 하드웨어 환경

고, 각 디스크 큐에 대기중인 요구의 스케쥴링, SCSI 어댑터로의 읽기 명령 전송, 통신 서브 시스템에 송신 명령 전송 등의 역할을 담당한다. 이러한 작업은 최소한 100ms 이상의 주기를 갖고 반복적으로 수행되며 한 주기내에 10~50개의 요구를 고려할때 CPU는 멀티미디어 서버에서 병목현상을 발생시키지 않는다. 또한 PCI 버스도 40MB/s이상의 유효 전송율(effective transfer rate)을 가지므로 10MB/s 전송율의 SCSI-II 버스를 고려할때 병목을 유발하지 않는다. 그리고 메모리도 디스크 서브 시스템의 성능을 최대한으로 발휘할 수 있을 만큼 충분하다고 가정하므로 본 논문에서 고려하는 멀티미디어 서버의 병목현상은 디스크 서브 시스템에서 발생한다고 할 수 있다.

2.2 저장구조

디스크 서브 시스템은 2~3MB/s의 낮은 대역폭을 갖는 디스크가 성능에 가장 큰 제약이 되므로 디스크 배열은 여러 디스크를 동시에 구동시킴으로써 이를 해결하고자 하는 것이다. 그런데 10MB/s의 대역폭을 갖는 SCSI-II 버스를 고려하면 5개 이상의 디스크를 사용할때에는 SCSI 버스가 병목을 유발하므로 여기에서는 4개까지의 디스크를 장착시킨 디스크 배열을 고려하기로 한다. 한편, 멀티미디어 서버의 디스크 서브 시스템으로 별도의 디스크 배열 어댑터(혹은 RAID adaptor)를 장착할 수 있으나 2.1에서 언급한 바와 같이 CPU의 부담이 없는 상황에서는 응용에 보다 적합하고 융통성있는 디스크 배열을 구성할 수 있는 소프트웨어적인 접근방법이 바람직하다.

디스크 배열은 크게 디스크들간의 병렬성(parallelism)을 이용한 구성(이하 병렬성 기법)과 병행성(concurrency)을 이용한 구성(이하 병행성 기법)으로 나눌 수 있다[1][1]. 병렬성 기법은 하나의 사용자 요구를 모든 디스크가 동기화되어 처리하고(RAID3), 병행성 기법은 여러 사용자 요구를 각 디스크가 독립적으로 서비스한다(RAID5). 병행성 기법은 소프트웨어적으로 구현하는데 무리가 없지만 병렬성 기법에서는 모든 디스크를 동기화시키기가 매우 어렵다. 그런데, 멀티미디어 서버에서는 주기적으로 사용자 요청이 발생되므로 병렬성 기법에서도 이러한 주기를 단위로 동기화되면 된다. 즉, 하나의 사용자 요구가 여러 디스크에 분배되더라도 동시에 처리할 필요없이 해당 주기내에 지정된 버퍼에 읽혀지면 된다. 따라서 멀티미디어 서버에서는 병렬성 기법이 소프트웨어적으로 구현 가능하다.

[1] 디스크 배열의 신뢰도를 높이기 위한 여분의 디스크(redundancy)는 데이타 손상시 즉각적인 복구(on-line reconvery)를 위한것이므로 여기에서는 성능측면만을 언급한다.

멀티미디어 서버의 성능은 동시에 서비스할 수 있는 사용자의 수로 결정된다. 이는 2.1절에서 언급한 바와 같이 디스크 배열의 능력에 달려있다. 먼저 병행성 기법에 대해 서비스 가능한 사용자 수를 분석해 보면 다음과 같다. 멀티미디어 스트림은 각 디스크에 스트라이핑(striping, 혹은 인터리빙, interleaving)되어 저장되어 있고 사용자가 요구하는 블럭 크기는 스트라이핑 크기와 같다. 따라서 한 디스크에서는 한 사용자에 대해 디스크 갯수 만큼의 주기마다 한 블럭씩을 서비스하면 된다. 이를 위한 조건식은

$$T_{seek\_max} + n \times \frac{s_{concurrent}}{R} \leq m \times T_{play}^{concurrent} \qquad (1)$$

이다. 여기에서 n은 사용자 수, m은 디스크 갯수, R은 디스크의 전송율, s는 블럭 크기를 각각 나타낸다. 그리고 T_{play}는 한 블럭이 재생되는 시간을 나타내며 사용자 요청의 주기가 된다. 또한 T_{seek_max}는 한 주기동안 n개의 요청에 대한 탐색시간 및 회전지연 등 기타 오버헤드를 포함한 값으로 SCAN 디스크 스케쥴링에 의해 최적화 될 수 있다[7]. 그런데 최근의 디스크는 탐색시간이 탐색거리의 제곱근에 비례하므로 T_{seek_max}를 설정하기가 어렵다. 식 (1)로부터

$$n \leq \frac{T_{play}^{concurrent} \cdot m - T_{seek\_max}}{s_{concurrent}/R} \qquad (2)$$

이므로 $m = 4$, $s = 36$KB, $R = 2.4$MB/s, MPEG-I 스트림을 가정하여 $T_{play} = 192$ms 일때 처리 가능한 영역은 그림 2의 빗금친 부분이다. 여기에서 평균 탐색시간을 15ms로 가정하면 T_{seek_max}

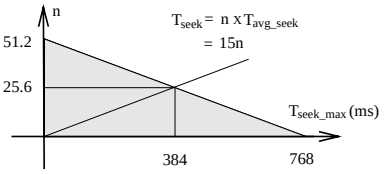


그림 2. n과 T_{seek_max}의 상관관계

는 그림 2에서와 같이 384ms이며 n은 25명이 된다.

병렬성 기법에서는 사용자가 요구하는 블럭의 크기가 스트라이핑 크기의 m배이므로 한 디스크에서는 한 주기동안 n/m개의 블럭, 즉, n개의 스트라이핑 유닛(unit)을 읽으면 된다.

$$T_{seek\_max} + \frac{n}{m} \times \frac{s_{parallel}}{R} \leq T_{play}^{parallel} \qquad (3)$$

그런데 스트라이핑 크기가 같을때 $s_{parallel} = m \cdot s_{concurrent}$이고, 따라서 $T_{play}^{parallel} = m \cdot T_{play}^{concurrent}$이므로 식 (1)과 (3)로부터 병행성 기법과 병렬성 기법의 성능이 같다는 것을 알 수 있다.

위의 분석에서 살펴보았듯이 디스크의 탐색시간과 성능간에는 밀접한 관계가 존재한다. 따라서 탐색시간의 비를 줄이기 위해서는 스트라이핑 크기가 가능한 커야한다. 그리고 스트라이핑이 트랙 단위로 이루어진다면 트랙 버퍼의 영향으로 회전지연시간이 없어지므로 스트라이핑 크기는 한 트랙 이상이 되어야 할 것으로 예상된다. 그러나 비용 대 성능 비를 고려할 때 적절한 스트라이핑 크기의 유도가 필요하며 디스크 내에서의 블럭 배치, 디스크 스케쥴링 등도 탐색시간에 영향을 미치는 요소이다. 다음절에서 이에 대한 실험을 수행한다.

표 1. 실험에 사용된 디스크 파라미터

Capacity	1.3 GB
Cylinders	1,962
Tracks per cylinder	19
Track size	36 KB
Revolution speed	4,002 RPM
Seek time	$3.24 + 0.400\sqrt{d}$ $(0 < d \leq 383)$
	$8.00 + 0.008d$ $(383 < d \leq 1962)$
Controller overhead	2.2 ms
Track switch time	1.6 ms

3 성능 분석

3.1 실험 모델

그림 3은 본 논문의 실험 모델을 나타낸다. 블럭 크기에 의한 주기마다 사용자 요청이 발생되고 사용자 요청은 디스크 배열의 구성 방법에 따라 각 디스크로 분배된다. 각 디스크는 해당 큐에 대해 탐색시간을 최적화하는 SCAN 디스크 스케쥴링을 한 후 읽기를 수행한다[2]. 멀티미디어 스트림의 연속성이 만족되는지 여부(해당 주기 내에 모든 블럭을 검사)를 검사하며 사용자의 수를 증가시켜 최대한의 서비스 가능한 사용자의 수를 구한다. 1.5Mbps의 MPEG-I 스트림을 가정하였으며 실험에 사용한 디스크의 파라미터는 표 1과 같다[5].

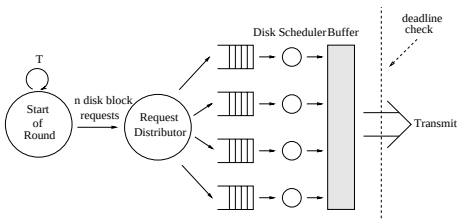


그림 3. 실험 모델

3.2 실험 결과

그림 4(a)는 디스크의 갯수가 4일때 스트라이핑 크기의 변화에 대해 각 디스크 배열의 구성에 따른 서비스 가능한 사용자의 최대수를 보여준다. 병행성 기법, 병렬성 기법 및 이의 혼합(hybrid) 기법에서는 스트림이 스트라이핑 크기로 각 디스크에 분산 저장되어 있다. 또한 스트림이 분산되지 않고 특정 디스크에만 저장되어 있는 기법(이하 단일배치 기법, no striping)에 대해서도 실험하였다[3]. 우선, 3절에서 분석한대로 병행성 기법, 병렬성 기법, 혼합 기법에 대한 차이는 크게 나타나지 않았다. 그런데 분석된 수치와 실험 결과는 상당한 차이를 보이는데 이는 T_{seek_max}의 과대설정에 기인한다. 병행성 기법과 병렬성 기법이 비슷한 성능을 보이는 상황에서는 블럭의 크기가 작은 병행성 기법이 우수하다고 할 수 있다. 이는 사용자의 초기 지연시간과 직접적인 관련이 있으며 사용자 요구의 QoS가 다를 경우 효율적인 스케쥴링으로 보다 많은 사용자를 서비스 할 수 있다[7].

단일배치 기법에서는 사용자 요구의 부하 분배에 따라 큰 차이를 보인다. 저장된 멀티미디어 스트림을 무작위적으로 선택한 경우는 어느 한 디스크가 병목을 유발하므로 문제가 발생된다. 인위적으로 각 디스크의 부하 분배를 균일하게 한 경우는 블럭 크기가 스트라이핑 크기보다 크므로 많은 사용자를 서비스 할 수 있다. 그러나 이러한 상황은 발생하기 힘드므로 단일 기법은 멀티미디어 서버 응용에 적합하지 않다고 볼 수 있다.

스트라이핑 크기의 변화에 대한 성능은 2.2절에서 언급한 바와 같이 한 트랙이상 할당시 월등한 향상을 보였다. 이는 탐색시간등의 오버헤드 영향이 상대적으로 줄어든 결과로서 그림 4(b)에 의해 알 수 있다[4]. 스트라이핑 크기의 증가로 인한 버퍼의 크기를 고려할 때, 최적의 스트라이핑 크기는 1~2 트랙으로 사료된다.

그림 4(c)는 디스크 갯수의 증가에 따른 선형적 성능 향상을 나타낸다. 그러나 5개 이상의 디스크일때는 SCSI 버스의 대역폭으로 인해 선형적인 성능 향상은 기대하기 어렵다. 따라서 어댑터당 4개의 디스크를 장착하는 디스크 배열이 적합함을 알 수 있다.

마지막으로 디스크 내에서의 블럭 배치가 성능에 미치는 영향을 실험하였다. 연속(contiguous) 배치는 임의(random) 배치에 비해 구현하기 쉽고 스트림을 관리하기 위한 메타 데이타가 필요없다는 장점을 갖는다. 그러나 스트림의 부분적인 변경이 많이 발생할 경우 디스크 공간의 단편화가 발생할 우려가 있다. 그림 4(d)에 의하면 두 배치 정책의 성능 차이는 거의 없다. 그런데 연속 배치는 디스크 캐쉬에 의한 성능 향상의 가능성을 갖는다. 그러나 디스크 캐쉬의 크기(128KB)를 고려할때 한 주기당 10개 이상의 블럭을 읽을 경우(360KB 이상) 디스크 캐쉬의 영향을 기대하기는 어렵다. 이는 실제 구현을 통해 확인해야 할 사항이다.

이상을 종합하여 본 논문에서 가정한 멀티미디어 서버의 저장 구조를 제안하면 표 2와 같다.

표 2. 본 논문에서 제안하는 저장 구조

디스크 배열 구성	병행성 기법
디스크 갯수	4
디스크 스케쥴링	SCAN
스트라이핑 크기	1~2 트랙
블럭 배치	연속배치

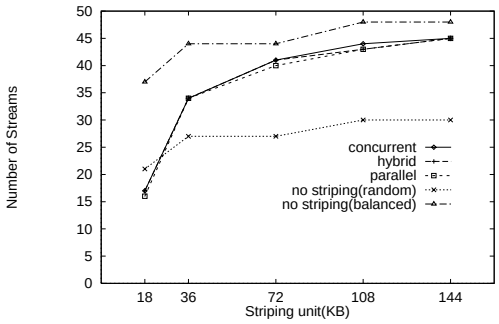
4 결론 및 향후과제

본 논문에서는 멀티미디어 서버를 위한 저장구조로서 디스크 배열에 대한 성능을 분석하고 실험을 통해 이를 확인하였다. PC를 기반으로 기존의 SCSI 어댑터와 SCSI 디스크로 디스크 배열을 소프트웨어적으로 구성할 경우, 하나의 어댑터에 4개의 디스크를 연결하여 멀티미디어 스트림을 1~2 트랙을 단위로 스트라이핑시키고 디스

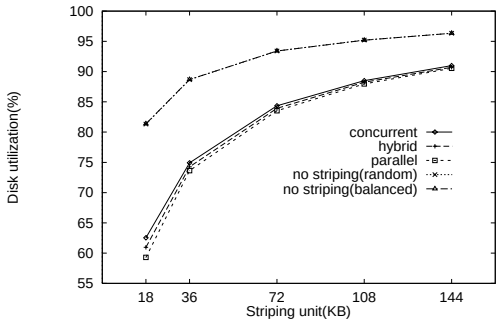
[2] 멀티미디어 서버는 WORM(Write Once Read Many) 응용이므로 읽기에 대해서만 고려한다.

[3] 그림 4에서 단일배치 기법의 블럭 크기는 (스트라이핑 크기× 디스크 갯수)로 보면된다.

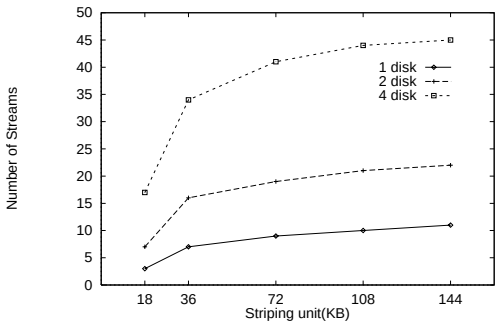
[4] 그림 4(b)에서 디스크 사용율은 (데이타 전송 시간/디스크 가동 시간)으로 정의된다.



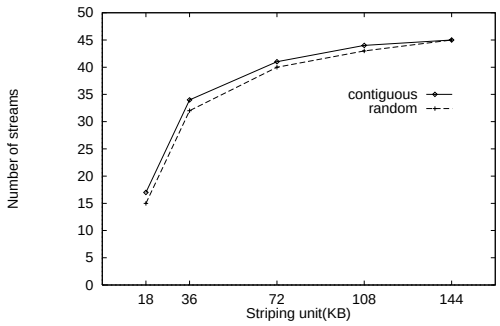
(a) 스트라이핑 크기 대 사용자 수



(b) 스트라이핑 크기 대 디스크 사용율 ($n = 15$)



(c) 디스크 수에 따른 성능 비교 (병행성 기법, $s = 36$KB)



(d) 블럭 배치 비교 (병행성 기법, $s = 36$KB)

그림 4. 실험 결과

크 내에서 연속배치로 저장한 후, SCAN 디스크 스케쥴링을 통한 병행성 기법을 이용하는 것이 가장 우수한 성능을 발휘하는 것으로 드러났다.

현재 구현중인 실험대(testbed) 상에서의 발생하는 문제에 대한 고려, 실험대 상에서 실험을 통한 확인, PC를 기반으로한 작은 규모의 서버를 서로 연동하여 수십~수백명의 사용자를 서비스할 수 있는 구조의 설계 등이 앞으로 수행해야 할 연구과제이다.

참고 문헌

[1] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," *ACM SIGMOD*, pp. 109–116, Jun. 1988.

[2] P. Lougher and D. Shepherd, "The design and implementation of a continuous media storage server," *Proc. 3rd International Workshop on Network and OS Support for Digital Audio and Video*, pp. 63–74, 1992.

[3] F. A. Tobagi, J. Pang, R. Baird, and M. Gang, "Streaming RAID^{TM} – A disk array management for video files," *Proc. 1st ACM International Conference on Multimedia*, pp. 393–400, 1993.

[4] K. D. Huynh and T. M. Khoshgoftaar, "Performance analysis of advanced I/O architectures for PC-based video servers," *Multimedia Systems*, Vol. 2, No. 1, pp. 36–50, 1994.

[5] C. Ruemmler and J. Wilkes, "An introduction to disk drive modeling," *IEEE Computer*, Vol. 27, No. 3, pp. 17–28, 1994.

[6] 서매실, 김치하 , "디스크 배열에서의 멀티미디어 데이타 할당 방법," 한국정보과학회, 제21권, 제5호, pp. 887–899, 1994.

[7] 조진성, 신현식 , "디스크 배열을 이용한 실시간 멀티미디어 저장 서버에서의 스케쥴링 기법," 한국정보과학회, 제21권, 제7호, pp. 1981–1989, 1994.