

다중 DB환경에서의 디스크 접근 속도 향상을 위한 하이브리드 스트라이핑 기법*

오창석^o 임성화 김재훈 고영배
 아주대학교 정보통신전문대학원
 {winstone^o, holyfire, jaikim, youngko}@ajou.ac.kr

Hybrid Striping Scheme for Improving Disk Access Time in Multi-DB

Chang-seok Oh^o Seong-Hwa Lim Jai-Hoon Kim Young-Bae Ko
 Graduate School of Information and Communication, Ajou University

요 약

스트라이핑(striping) 기법은 여러 개의 디스크에 데이터를 병렬적으로 분산하여 저장함으로써 데이터 접근 속도를 높이기 위하여 제안되었다. 그러나 서로 다른 데이터를 요청하는 다수의 사용자 요구 환경에서 스트라이핑 기법은 seektime과 rotational latency의 증가로 인해, 오히려 I/O 성능이 저하될 수 있다. 본 논문에서는 서로 다른 데이터 요구율을 갖는 다중 데이터베이스 액세스 환경에서 스트라이핑 폭(width)에 의한 입출력 성능을 분석하고, 사용자 요구 형태에 따라 스트라이핑 디스크 그룹과 개별 디스크 그룹으로 디스크 풀(pool)을 나누어 처리하는 하이브리드 스트라이핑 방식을 제안한다. 제안 방식의 성능 향상은 시뮬레이션에 의하여 검증하였다.

1. 서 론

데이터베이스를 기반으로 동작하는 많은 프로그램에서 디스크 액세스 시간은 CPU에 의한 계산 시간보다 상대적으로 큰 비중을 차지한다. VOD와 같은 스트리밍(streaming) 서비스가 그 대표적인 예라 할 수 있다. 이러한 프로그램에서 디스크 풀(pool)의 설정은 I/O 성능에 큰 영향을 미친다.

데이터 스트라이핑(striping) 기법은 디스크 I/O 속도 향상을 위한 방안으로 데이터를 병렬적으로 분산하여 디스크에 저장함으로써 사용자의 데이터 요구시, 각 디스크의 병렬적인 읽기 능력에 의한 빠른 데이터 전송속도를 얻는 방식이다[1]. 이는 이미 RAID(Redundant Arrays of Inexpensive Disks)와 같은 대용량 저장매체에서 사용하고 있는 방식으로 부하분산을 통해 부하가 특정 디스크에 집중되는 핫스팟(hot spot) 문제를 해결한다. 그러나 처음부터 사용자 요청이 고르게 분산되어 들어온다고 가정할 때, 핫스팟 문제는 발생하지 않고, 이때 I/O 성능을 좌우하는 주된 요소는 동시에 처리할 수 있는 사용자 요구수라 할 수 있다. 스트라이핑 기법은 한명의 사용자 요구를 처리하는 동안 다른 사용자들은 대기 상태에 있어야 하므로 사용자의 요구수가 많은 경우 seek time과 rotational latency가 기존의 기법보다 증가하는 단점이 있다.

본 논문에서는 사용자 요구가 일부 데이터에 편중된 데이터베이스와 고른 분포를 갖는 데이터베이스가 함께 공존하는 시스템에서 디스크 액세스 시간을 최소화하기 위한 방안으로 하이브리드 스트라이핑 방식을 제안한다. 하이브리드 스트라이핑 방식은 편중된 데이터 액세스에 의해 발생하는 핫스팟 문제를 디스크 풀(pool) 중 일부 디스크에 스트라이핑함으로써 부하를 분산하고, 나머지

디스크는 개별적으로 유지함으로써 동시에 액세스가 가능한 디스크 수를 확장하였다. 이러한 응용은 생물 정보학 서비스에서 각종 유전자 데이터베이스를 검색하고 처리할 때 데이터베이스 종류별로 또는 유전자 종류별로 검색빈도가 다를 수 있기 때문에 이를 동시에 효과적으로 서비스 할 수 있는 방법이 될 수 있다. 또한 인기도에 따라 사용자 요구를 차이가 큰 최신 비디오와 고른 사용자 요구율을 갖는 클래식 비디오 데이터를 동시에 서비스 하기위한 VOD서버에서 디스크 I/O의 성능을 개선할 수 있는 방법이 될 수 있다.

2. 성능분석

2.1 스트라이핑 폭(striping width) 변화에 따른 성능 분석

I/O 성능에 있어서 성능 결정의 주요변수는 스트라이핑 폭(striping width)이다. 스트라이핑 폭은 몇 개의 드라이브에 데이터를 병렬적으로 저장할 것인지를 결정하는 요소로 데이터 전송속도는 스트라이핑 된 드라이브 개수에 반비례한다. 본 연구에서는 [그림 1]과 같이 8개의 드라이브를 모델로 스트라이핑을 사용하지 않았을 경우(non-striping), 스트라이핑 폭을 4로 하였을 경우(partial-striping), 스트라이핑 폭을 8로 하였을 경우(full-striping), 마지막으로 우리가 제안한 방식인 폭을 4로 스트라이핑한 그룹과 스트라이핑 하지 않은 그룹을 혼합한 하이브리드 스트라이핑(hybrid striping) 모델에 대해 분석하였다.

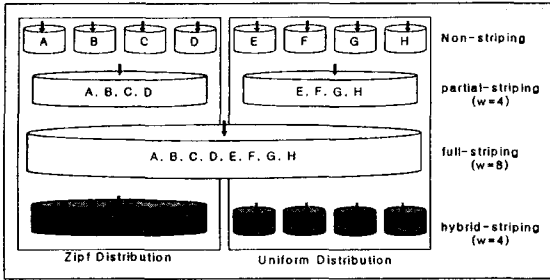
모델 분석에 사용한 파라미터는 [2]에서 참조한 [표 1]과 같다.

[표 1] 성능분석에 사용된 파라미터 [2]

parameter	notation	normal value
disk block size	s	1Mbyte
disk transfer rate	r	4Mbyte/sec
seek time	t_s	10msec
rotational latency	t_r	6msec
striping width	w	1 - 8
number of disks in array	D	8

* 본 연구는 정보통신부 정보통신선도기반기술개발사업의 지원에 의하여 이루어진 것임.

데이터 액세스 시간은 디스크 오버헤드인 seek time과 rotational latency, 그리고 데이터 전송시간의 합으로 정의된다.



[그림 1] Disk pool setup

스트라이핑을 사용하지 않았을 경우 T초간 서비스 할 수 있는 블록 수는 모든 사용자 요청이 하나의 드라이브로 집중되었을 경우(C_{min}), 즉 핫스팟 문제가 발생했을 경우와 서로 다른 드라이브에 저장된 데이터에 액세스 하려 할 경우(C_{max})로 나뉜다.

$$C_{max} = D * \frac{T}{l_s + l_r + \frac{s}{r}}, \quad C_{min} = \frac{T}{l_s + l_r + \frac{s}{r}}$$

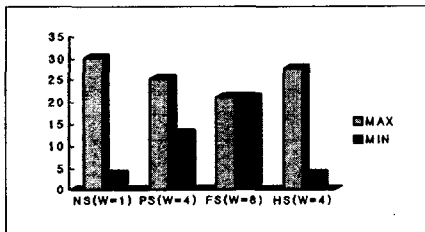
스트라이핑 폭을 w 로 하였을 경우 T초간 서비스 할 수 있는 블록 수는 하나의 스트라이핑 그룹이 서비스 할 수 있는 블록 수와 동시에 서비스 할 수 있는 사용자 요구 수, 즉 스트라이핑 그룹의 개수와의 곱으로 나타낼 수 있다.

$$C_{max} = \frac{D}{w} * \frac{T}{l_s + l_r + \frac{s}{rw}}, \quad C_{min} = \frac{T}{l_s + l_r + \frac{s}{rw}}$$

예를 들어 우리가 분석하려는 모델인 partial-striping의 경우 스트라이핑 그룹은 2개가 되고, full-striping의 경우엔 1개의 스트라이핑 그룹을 갖는다. 하이브리드 스트라이핑 방식은 1개의 스트라이핑 그룹과 4개의 개별 드라이브로 구성된다. 따라서 T초간 서비스 할 수 있는 블록 수는 사용자 요구가 편중되지 않을 경우,

$$C_{max} = \frac{T}{l_s + l_r + \frac{s}{rw}} + (D-w) * \frac{T}{l_s + l_r + \frac{s}{r}}$$

만공의 블록을 서비스 할 수 있고, 스트라이핑 되지 않은 데이터에 사용자 요구가 집중될 경우 스트라이핑을 사용하지 않았을 경우와 같은 성능을 나타낸다. 그림[2]은 1초간 서비스 할 수 있는 블록 수를 나타낸 그림이다.



[그림 2] 1초간 서비스 할 수 있는 블록 수

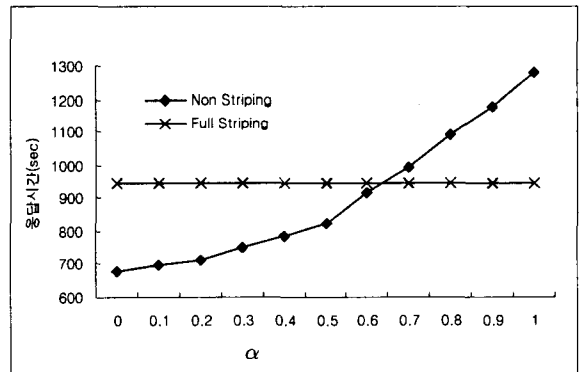
2.2 사용자 요구 편중에 따른 응답시간

사용자 요구 편중에 따른 각 스트라이핑 모델별 응답시간은 zipf 분포에서 α 값 변화에 따라 성능의 차이를 보인다. zipf 분포는 액세스 빈도 순위(i)에 의해 사용자 요구가 들어왔을 때 각 디스크별 액세스할 확률($z(i)$)을 나타낸다.

$$\text{Zipf Distribution} : C = 1 / \sum_{k=1}^D (1/k^\alpha), \quad Z(i) = C/i^\alpha$$

α 값이 0일 때 사용자 요구 분포는 uniform 분포를 나타내어 사용자 요구가 고른 분포를 나타내고, α 값이 증가할수록 인기도 차이가 큰 최신 비디오특정 데이터로의 편중 현상은 심해진다.

[그림 3]은 α 값의 변화에 따른 non striping 기법과 full striping 기법의 응답시간의 변화를 보여준다. Full striping의 경우, 특정 데이터로의 사용자 요구 편중에 상관없이 일정한 응답 시간을 나타내고 있다. 이는 스트라이핑에 의해 각 디스크로 부하가 분산된 결과이다. 이와는 대조적으로 요구 데이터 편중이 심해질수록 non striping 모델은 응답시간이 급격히 증가하였다. 이는 특정 디스크로 부하가 집중됨으로 인한 핫스팟 문제가 발생하기 때문이다. 사용자 요구가 고르게 발생할 경우 ($\alpha = 0$), 스트라이핑을 사용하지 않았을 때의 응답시간이 스트라이핑을 사용했을 때보다 작았다. 이는 동시에 처리할 수 있는 사용자 요구수가 스트라이핑 하였을 경우보다 많으므로 seek time과 rotational latency가 감소하기 때문이다.

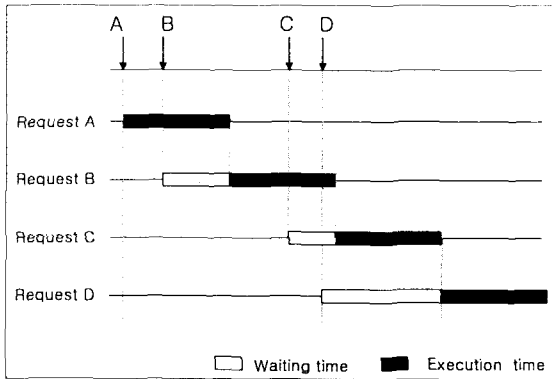


[그림 3] zipf distribution에서 α 의 변화에 따른 응답시간 변화 [총 사용자 요구수 = 10000회, 사용자 도착률 = 1000회/sec]

3. 하이브리드 스트라이핑 기법의 성능평가(시뮬레이션)

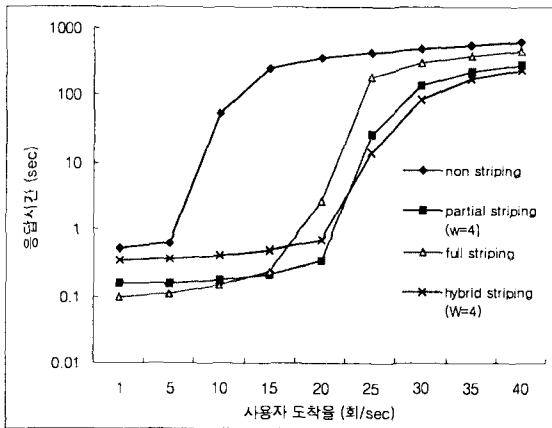
본 연구에서 시스템은 사용자의 데이터 요구 경향이 특정 데이터에 편중된 데이터베이스(최신비디오 DB)와 고른 분포를 갖는 데이터베이스(클래식 비디오 DB) 모두를 포함하고 있다고 가정한다. 또한 양 데이터베이스의 액세스 회수는 동일하다고 가정한다. [그림 4]는 동일한 드라이브의 데이터에 대한 사용자 요구의 응답시간을 나타낸다. 사용자 요구가 들어왔을 때, 현재 액세스 하고 있는 다른 사용자가 없을 경우 데이터 액세스는 바로 이루어지고, 이미 다른 데이터 액세스가 일어나고 있는 경우 이전의 데이터 액세스가 끝날 때 까지 기다린후 자신의 요청이 서비스 된다. 시뮬레이션은 [표 1]의 파라미터를 사용하였고, 데이터 편중을 위한 zipf 분포의 α 인자는 일반적으로 VOD 서비스에서 사용되는 지수인 2[5]와 uniform 분포를 위한

인자 0을 사용하였다. 최대 10000회의 사용자 요구를 Poisson process로 발생시켰고, 평균 응답시간은 전체 I/O시간을 사용자 요구수로 나누어 계산하였다. 사용자가 요구하는 평균 데이터 크기는 입출력 단위인 1블록을 기준으로 계산하였다.



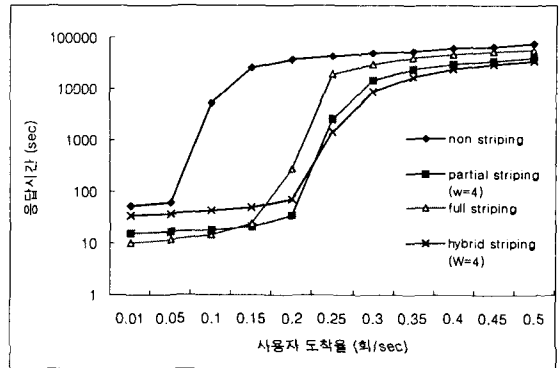
[그림 4] 사용자 요구 처리를 위한 디스크 I/O

[그림 5]는 초당 평균 사용자 도착률에 의한 각 모델별 응답시간의 변화를 나타낸다. 초당 사용자 도착률이 낮을 때, full striping의 성능이 가장 좋을 수 있다. 이는 Full striping 모델의 병렬 입출력 능력으로 인해 하나의 사용자 데이터 요구에 대한 처리 시간이 다른 모델보다 상대적으로 우수함을 나타낸다. 또한 non striping 모델은 평균 도착률이 5 부문에서 핫스팟 문제가 발생하였음을 볼 수 있다. 하이브리드 스트라이핑 방식과 full striping 방식의 응답시간은 15와 20 사이 지점에서 역전되고, 스트라이핑 폭이 4로 설정된 partial striping 방식과의 비교에서도 사용자 도착률이 25이후 지점에서 하이브리드 스트라이핑 방식이 보다 우수함을 볼 수 있다.



[그림 5] 사용자 도착률 변화에 따른 각 모델별 응답시간 변화
[$\alpha=2$, 총 사용자 요구수=10000회, 데이터 크기=1블록]

즉, 하이브리드 스트라이핑 방식은 초당 사용자 도착률이 증가할수록 다른 방식보다 우수한 성능을 보인다. 뿐만 아니라, 사용자가 요구하는 평균 데이터 크기를 1블록에서 100블록으로 증가시킬 경우 사용자 도착률이 0.25 이후 지점부터 하이브리드 스트라이핑 방식이 우수한 성능을 보였다.



[그림 6] 큰 데이터 요구시 각 모델별 응답시간 변화
[$\alpha=2$, 총 사용자 요구수=10000회, 데이터 크기=100블록]

이러한 시뮬레이션의 결과는 특정 데이터에 대한 사용자 요구 편중 정도가 서로 다른 다중 DB환경에서 하이브리드 스트라이핑 방식이 스트라이핑을 전혀 사용하지 않거나(non striping), 확실적인 스트라이핑 전략을 이용할 때(full striping)보다 더 효율적임을 나타낸다.

4. 결론

스트라이핑 방식은 병렬적인 입출력 능력에 의한 전송속도의 개선으로 한 사용자 요구에 대한 처리속도를 향상시킨다. 또한 부하 분산을 통해 핫스팟 문제를 해결한다. 그러나 사용자 요청이 증가함에 따라 기존 기법에 비해 응답시간이 급격히 증가하는 단점을 갖는다. 따라서 DB의 성격에 따른 사용자 요구 경향을 사전에 미리 판단하여 적절한 스트라이핑 전략을 수립하는 것이 중요하다. 본 논문에서는 특정 데이터에 대한 사용자 요구 경향이 서로 다른 다중 DB 환경에서 디스크 풀(pool)을 스트라이핑 그룹과 개별 디스크 그룹으로 나누어 설정함으로써 핫스팟 문제를 해결하고, 동시에 서비스 할 수 있는 사용자 요구수를 최대한 보장하는 하이브리드 스트라이핑 방식을 제안하였다. 초당 사용자 요구수가 증가할수록 하이브리드 스트라이핑 방식은 다른 모델에 비해 빠른 응답시간을 나타낸다. 다중 DB 환경에서 하이브리드 스트라이핑 방식은 데이터 복제없이 디스크 용량을 최대한 활용하면서 디스크 I/O 효율을 높이는 큰 효과를 나타낸다.

참고문헌

- [1] P. M. Chen, G. A. Gibson, D. A. Patterson, E. K. Lee, and R. H. Katz, "RAID : High, Reliable Secondary Storage," ACM Computing Surveys, Vol. 26, No. 2, pp. 145-185, June 1994.
- [2] M. Reisslein, K. W. Ross, and S. Shrestha, "Striping for Interactive Video : Is it worth it?," Computing and Systems, Vol. II-Vol. 2, pp. 635-640, June 1999.
- [3] P. Triantafyllou, and C. Faloutsos, "Overlay Striping and Optimal Parallel I/O for Modern Application," Parallel computing, Vol. 24, pp.21-43, January 1998.
- [4] S. Chen, and D. Towsley, "A Performance Evaluation of RAID Architecture," IEEE Transactions on computers, pp. 1116-1130, 1996.
- [5] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, "Web Caching and Zipf-like Distribution : Evidence and Implications," IEEE INFOCOM, The Conference on Computer Communications, No. 1, pp. 126-134, March 1999.