로그구조 디스크 서버에서 부분 세그먼트의 영향

김대웅○ 남영진 박찬익
포항공과대학교 전자계산학과

Partial Segement Effects on the Log-structured Disk Server

Dae-Woong Kim Young-Jin Nam Chan-Ik Park
Dept. of Computer Science and Engineering/PIRL, POSTECH

요 약

본 논문에서는 기존의 로그구조 디스크 서버(Log-structured Disk Server; LDS) 상에서 부분 세그먼트에 의한 디스크 대역폭 이용률 저하 및 높은 공간 정리 부하 (cleaning overhead)량과 같은 문제점을 제기하고, 이러한 부분 세그먼트를 제거하기 위한 방법의 하나로 비휘발성 메모리를 사용한 LDS를 제안한다. 이 새로운 LDS를 Solaris 운영체제 상에서 구현하였으며, 다양한 실험을 통하여 제안된 구조가 디스크 대역폭 이용률 및 공간 정리 부하 측면에서 기존의 구조에 비해서 월등히 좋은 성능을 보임을 볼 수 있었다.

1. 서론

파일 입출력 시에 발생하는 다수의 작은 쓰기(Small write)로 인한 물리적 디스크로의 빈번한 접근 회수를 줄이기 위한 접근 방법으로 [1, 2] 등이 제안되었다. 이 중에서 로그구조 저장 장치 (*LDS: Log-Structured Disk Server*)는 쓰고자하는 작은 데이터들을 세그먼트라는 구조체에 모은 뒤에 그 세그먼트를 디스크에 한번에 씀으로써 디스크 접근시간을 줄이고자 하였다 [2]. 하지만, 쓰기 지연시간(write delay time) 초과, 응용 프로그램에서 발생한 fsync 등으로 인해서 세그먼트가 완전히 채워지기 전에 디스크에 써야만 하는 경우가 종종 발생한다. 이렇게 생긴 세그먼트를 부분 세그먼트 (partial segment)라고 부른다. 부분 세그먼트가 빈번히 발생하면 그만큼 디스크 동작이 많아지므로 성능에 큰 영향을 줄 수 있다. 또한 빈 세그멘트를 빨리 소모하므로 공간정리(cleaning) 작업에 의한 부하가 커진다.

쓰기지연 시간에 한계를 두는 이유는 휘발성 메모리(volatile memory)로 구성된 캐쉬에 저장되어있는 더티(*dirty*) 데이타는 소실될 위험이 있기 때문이다. 따라서 되도록 빨리 비휘발성 메모리(nonvolatile memory, NVRAM)나 디스크 등의 비휘발성 저장장치에 이동되어야한다. 반면 캐쉬 쓰기 적중률(cache write hit rate)을 높이려면 더티 데이터는 가능한 오랫 동안 캐쉬에 머물러야한다. 디스크로의 읽기 작업은 캐쉬를 크게 두어 효과적으로 줄일 수 있다. 하지만 디스크로의 쓰기 작업은 쓰기 지연시간의 제약 때문에 읽기 작업에 비해 줄이기가 힘들다 [1].

본 논문에서는 위와 같은 원인들에 의한 부분 세그먼트 발생을 피하기 위한 방법의 하나로 비휘발성 메모리를 사용한 LDS를 제안한다. 이 구조에서는 쓰기 지연시간의 제약을 제거할 수 있어 디스크로의 쓰기 작업을 크게 줄일 수 있다.

2. 제안된 로그구조 디스크 서버

본 논문에서 제안된 로그 구조 서버는 Logical Disk [3]의 구현 중 하나인 Log-structured Logical Disk(LLD)에 비휘발성 메모리가 포함된 구조를 가진다.

LDS는 Sprite-LFS [2]나 BSD-LFS [4]와 같은 기존의 파일 시스템과 달리 블록의 읽기, 쓰기 등의 디스크와 비슷한 인터페이스를 가진다. LDS는 파일시스템의 버퍼 캐쉬와는 별개로 NV cache(비휘발성 메모리 캐쉬)를 둘 수 있다. 그림 1은 LDS 시스템의 주요 동작을 보여준다.

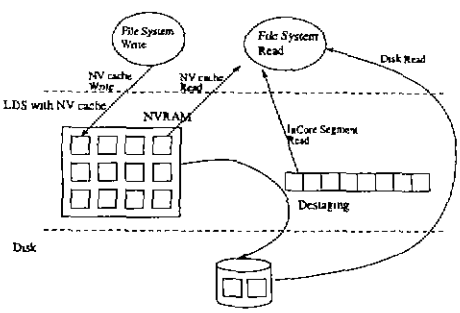


그림 1: LDS가 NV cache를 사용했을 때, 주요 동작과 데이터 블록의 이동

NV cache가 어느 정도 차면 디스크에 저장하기 직전에 더티 데이터와 메타 데이터로 세그먼트를 휘발성 메모리 상에 구성한다. 세그먼트가 디스크에 쓰여지면 블록 주소표에서 해당 블록의 물리적 주소, 수정 시간 등의 정보를 바꾸어준다. 블록주소표는 논리적 블록 번호를 물리적 디스크 주소로 사상(mapping)하기 위한 정보 등을 가지는 구조체이다.

파일시스템이 특정 블록을 읽는다는 요청을 보내면 LDS는 그 블록을 먼저 휘발성 메모리상의 세그먼트에서 찾고, 만약 없으면 NV cache에서 찾는다. LDS 버퍼들에서 해당 블록을 찾지 못하면 블록주소표에 따라 위치를 알아낸 다음 디스크를 읽어 요구된 데이터 블록을 읽는다.

본 실험에서 사용한 디스테이징(destaging) 알고리듬은 경계치(threshold) 기법과 비슷하다. 파일 시스템 등의 요구를 처리하면서 NV cache내에 더티 블록이 증가한다(a). NV cache에 더티 데이터가 *경계치(threshold)*를 넘게되면(b) 세그먼트를 구성하여 디스크에 저장한다(c). 더 이상 자유(free) 블록이 없으면 앞서의 세그먼트 쓰기가 끝나기를 기다려 디스테이지 된 블록을 자유블록으로 설정한다(d).

공간정리 정책은 Sprite-LFS에서 제시한 정책 중 가장 간단한 방법을 선택했다. 공간 정리는 디스크 상에 빈 세그먼트가 2개 남게되면 시작된다. 공간정리 될 세그먼트로는 살아

Interface	Fast-20
Read/Write Heads	2
Number of Cylinders	4,162
Sectors/Track	126
Internal Data Rate	44 - 66 Mbits/sec
Disc Rotational Speed	5,411 ± 0.5 % r/min
Average Rotational Latency	5.54 ms
Typical Access Time (in ms)	10.4 (Read), 11 4 (Write)

표 1: Seagate ST32155N 디스크 드라이브의 특성

있는(live) 블록이 가장 작은 것을 선택한다. 또, 선택된 세그먼트는 한번의 디스크 읽기로 읽어들인다.

3. 성능 측정 및 분석

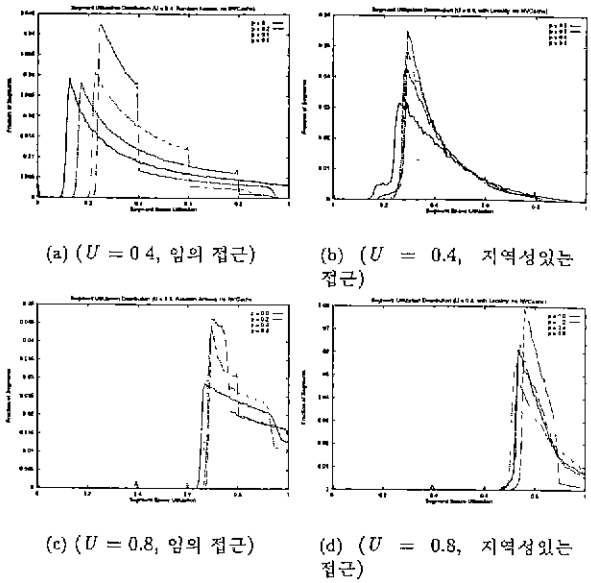
실험에 사용한 머신은 Sparc 20이며 32 Mbytes의 주기억장치를 가졌고, Solaris 2.5를 운영체제로 한다. 블록 크기는 4 Kbytes로, 세그먼트 크기는 128 블록으로 설정하였고, 2 Gbytes의 디스크 중 64,000 블록 크기만큼을 이용하였다. NV cache 크기는 1, 2, 4 Mbytes로 두었고 휘발성 메모리인 주기억장치에서 할당받아 비휘발성 메모리라 가정하였다. 또, 일반적으로 공간정리 비용은 총 디스크 공간 중 살아있는 블록이 차지하는 비율인 디스크 공간 이용율(disk space utilization; U)이 클수록 커진다. 따라서 이런 특성을 반영하기위해 U가 일정값 유지하는 평형상태가 되도록 입출력요구를 발생시켰으며, U 값으로 0.4와 0.8를 사용하였다. 사용한 디스크 모델은 'Seagate ST32155N'으로 주요 특성은 표 1과 같다.

입출력 요구는 인공적으로 발생시켰으며 읽기 작업은 없고 오직 갱신 작업으로만 이루어져있다. 접근 대상은 임의 접근(random access)하는 것과 자주 접근되는 지역(hot spot)을 설정하여 지역성을 고려한 두가지를 실험하였다. 지역성을 위해 A, B, C, D의 4개의 영역을 두었는데 각각의 크기와 접근빈도를 달리두었다. 즉, A 영역의 크기는 512 블록이고 총 요구중 50 %가 접근한다. 마찬가지로, B 영역은 512 블록, 30 %, C영역은 1024 블록, 10 % 그리고 D 영역은 나머지 블록으로 이루어져있으며 총 요구의 10 %가 접근한다.

Sprite-LFS와 BSD-LFS에서는 하나의 세그먼트에 여러 개의 부분 세그먼트가 들어가는 방식을 사용했다. 그러면 각각의 부분 세그먼트마다 자신에 대한 메타정보인 요약 블록(summary block)을 두게된다 반면에 본 실험에서는 하나의 세그먼트에 많아야 하나의 부분 세그먼트가 들어가도록 했다. 즉, 세그먼트가 채워진 양에 관계없이 언제나 디스크 상에 새로운 세그먼트를 할당받는다. 이런 경우 세그먼트를 빨리 소모하게되어 공간정리 부하가 커지게 된다. 이러한 효과만 유의한다면, Sprite-LFS와 BSD-LFS의 방법을 쓰는 것보다 분석이 간단 명료하게된다.

부분 세그먼트가 성능에 끼치는 영향은 크게 두 가지로 볼 수 있다. 첫째, 디스크 요구 수가 증가한다. 따라서 디스크 탐색 시간에 의한 부하가 늘어난다. 둘째, 디스크 공간을 비효율적으로 사용함에 따라 공간정리 작업을 더 자주 요구하게 된다.

일단 시작된 공간정리작업의 효율성에 영향을 미치는 가장 큰 요소는 공간정리 작업이 발생한 시점에서 세그먼트 (공간) 이용율 (segment space utilization)의 분포이다. 세그먼트의 이용율은 한 세그먼트안의 살아있는 블록의 비율이다. 공간정리를 위해 선택된 세그먼트들의 이용율이 작으면 작을수록 적은 노력으로도 많은 빈 세그먼트들을 생성할 수 있기 때문이다. 세그먼트 이용율의 분포에 영향을 미칠 수 있는 요소 중 하나는 초기 세그먼트 이용율(u)로 세그먼트가 디스크에 쓰여질 당시의 이용율이다. 다른 요소로는 세그먼트의 평균 수명(Mean Life Time; MLT)을 들 수 있으며, 이는 한 세



(a) ($U = 0.4$, 임의 접근) (b) ($U = 0.4$, 지역성있는 접근)

(c) ($U = 0.8$, 임의 접근) (d) ($U = 0.8$, 지역성있는 접근)

그림 2: NV cache를 이용했을때, 부분 세그먼트 정도에 따른 세그먼트 이용율의 분포

그먼트가 디스크에서 쓰여진 시점에서 공간정리의 대상으로 선택되는 시점까지 기간동안 디스크 서버의 디스크에 쓰여진 블록 개수의 평균으로 정의한다. 세그먼트의 평균 수명은 부분 세그먼트의 발생빈도와 크기에 의해 영향을 받으며, 이를 구하기 위해서는 다음과 같은 식을 이용한다.

$$MLT = \frac{\text{세그먼트 쓰기 회수} \times \text{세그먼트당 블록의 평균 수}}{\text{공간 정리를 위한 세그먼트 읽기 회수}} \quad (1)$$

세그먼트 쓰기 회수는 클라이언트로부터의 쓰기 요구를 만족시키기 위해 수행한 세그먼트 쓰기의 회수를 말하고, 세그먼트당 블록 수는 세그먼트 내의 살아있는 블록의 평균 개수를 뜻한다. 예를 들어 세그먼트 크기가 128블록이고 $u = 0.4$이면 세그먼트당 블록의 평균 수는 대략 $51(= 128 \times 0.4)$이다. 공간 정리를 위한 세그먼트 읽기 회수는 공간 정리를 위해 선택된 세그먼트의 총수와 같다.

부분 세그먼트에 의한 영향을 살펴보기 위해서 인위적으로 부분 세그먼트를 발생시켰다. 표 2, 3은 임의 접근을 하고 각각의 디스크 공간이용율(U)을 0.4와 0.8로 고정한채 부분 세그먼트 정도(partialization degree)를 나타내는 p를 변화시켰을 때의 실험치들을 정리한 것이다. 여기서 p는 $1 - u$이다 앞서 예상했던대로 p가 커질 수록 주어진 사용자 요구를 만족시키는데 더 많은 세그먼트 쓰기를 한다. 또한 세그먼트를 빨리 소모하므로 공간정리 부하가 커진다. $U = 0.4$인 경우의 총 시간은, 각각 $p = 0.2$, $p = 0.4$, $p = 0.6$ 일 때, 부분 세그먼트가 없을 때($p = 0.0$)보다 1.344, 2.011, 3.049 배 정도 걸리며 $(1 - p)$에 반비례하는 경향을 보인다. 이는 $U = 0.8$일 때도 비슷하며 성능이 p에 좌우되는 것을 알 수 있다.

세그먼트 이용율의 분포는 p와 세그먼트의 평균 수명에 의해 결정된다. 표에서 세그먼트의 평균수명은 수식 1으로 구해진다. $U = 0.4$ 인 그림 2(a)를 보면 p가 작을 수록 최소 세그먼트 이용율이 작아지는데 그 이유는 초기 세그먼트 이용율이 큼에도 불구하고 세그먼트 평균수명이 길기때문이다. 반면에 $U = 0.8$인 경우, p에 따른 최소 세그먼트 이용율은 거의 비슷하였다. 이는 세그먼트 평균수명의 영향과 초기 세그

p	Real Workload		Cleaning			Total
	Num of Seg. W.	Time (sec)	Num of Seg. R.	Num of Seg. W.	Time (sec)	Mean Lifetime
0.0	5,087	564.3	657	6,380	900.7	112.04
0.2	6,399	711.2	1,273	8,911	1,259.0	84.70
0.4	8,509	941.8	2,810	14,051	2,005.2	57.15
0.6	12,941	1,414.9	4,274	21,370	3,052.4	38.10

표 2: 임의 접근, $U = 0.4$일때, 공간정리 부하 (NV cache 사용하지 않음)

p	Real Workload		Cleaning			Total
	Num of Seg. W.	Time (sec)	Num of Seg. R.	Num of Seg. W.	Time (sec)	Mean Lifetime
0.0	5,093	555.5	15,279	10,186	3,617.8	42.33
0.2	6,407	679.5	25,154	18,747	6,193.9	25.87
0.4	8,516	882.7	22,766	17,032	5,662.6	28.50
0.6	12,947	1,302.5	34,718	25,894	8,642.5	18.94

표 3: 임의 접근, $U = 0.8$일때, 공간정리 부하 (NV cache 사용하지 않음)

먼트 이용율의 영향이 상쇄되기때문이라 생각한다.

공간 정리 작업이 얼마나 효과적으로 수행되었는가는 사용자 요구처리 시간에 대한 공간정리 시간과의 비율로 알 수 있으며, 이값은 최소 세그먼트 이용율이 작을 수록 작은 것을 알 수 있다. 표 2, 3이 임의 접근인 경우와 달리 표 4는 네개의 크기가 다른 영역을 두고 각각에 접근하는 빈도수를 다르게 했을 때의 실험 값들이다. 지역성을 두었을 때의 각종 성능 수치들은 전체적으로 임의 접근과 비슷하였다. 그 이유는 버퍼캐쉬 역할을 하는 세그먼트의 크기가 아주 작아 디스크 쓰기 작업을 줄이는 효과가 거의 나타나지 않았기때문이다.

또한 일반적으로 디스크 공간 이용율은 시스템 성능에 큰 영향을 미친다. 그 이유는 디스크 공간 이용율은 세그먼트 이용율의 평균값이며 일반적으로 이값이 클 수록 최소 세그먼트 이용율도 커지기 때문이다.

표 5는 NV cache를 사용하고 임의 접근을 했을 때, $U = 0.4$인 경우의 디스크 쓰기 작업 감소 정도를 나타낸다. 표에서 NV cache의 크기는 1, 2, 4 Mbytes를 가진다. 한편 표 6, 7는 지역성을 갖는 접근을 했을 때의 성능 수치이다. 지역성을 갖지 않을 때는 NV cache가 커지더라도 성능향상의 정도는 미비하였다. 이에 반해 지역성을 갖을 때는 성능 향상이 두드러졌다. 캐쉬 쓰기 적중율로 그 현상을 설명할 수 있다. 지역성이 없는 경우 적중율은 0.17 %에서 3 %였다. 비록 NV cache가 커짐에 따라 적중율이 크게 향상되었지만 줄어든 디스크 쓰기 회수의 절대적인 수치가 매우 작아 그 효과가 거의 나타나지 않았다. 지역성을 가지는 경우, 적중율이 6 %에서 51 %로 NV cache가 커질 수록 디스크 쓰기 양을 크게 줄일 수 있다.

동일한 상황에서 지역성이 있는 경우와 그렇지 않은 경우 최소 세그먼트 이용율의 크기는 지역성이 없는 경우가 더 작았으며 따라서 공간 정리 부담도 작은 것을 알 수 있었다. 지역성이 있을 때 최소 세그먼트 이용율이 커지는 현상은 자주 접근되는 블록들은 캐쉬에서 만족되는 반면 그렇지않는 블록들은 결국 디스크에 쓰여지며 그 후로도 다시 접근될 확률이 작기때문이라 추측된다.

4. 결론 및 향후 연구 계획

본 논문에서는 기존의 로그구조 디스크 서버 상에서 부분 세그먼트에 의해 디스크 대역폭 이용률 저하 및 높은 공간 정리 비용이 발생함을 실험을 통해 살펴보았다. 부분 세그먼트 정도가 0.6일때는 부분 세그먼트가 생기지 않았을 때보다 지역성과 디스크 공간 이용율에 따라 2에서 3배 정도의 공간 정리 비용이 들어감을 알 수 있었다.

p	Real Workload		Cleaning			Total
	Num of Seg. W.	Time (sec)	Num of Seg. R.	Num of Seg. W.	Time (sec)	Mean Lifetime
0.0	4,878	536.5	19,195	14,317	2,109.8	32.27
0.2	6,192	652.0	24,622	18,430	2,680.4	25.55
0.4	8,300	862.3	21,896	17,498	3,583.6	28.88
0.6	12,730	1,306.8	32,645	25,604	5,402.0	19.80

표 4: 지역성있는 접근, $U = 0.8$일때, 공간정리 부하 (NV cache 사용하지 않음)

NV Cache Size	Real Workload		Cleaning			Total	
	Num of Seg. W.	Time (sec)	Num of Seg. R.	Num of Seg. W.	Time (sec)	Mean Lifetime	Hit Ratio
1 Mbytes	5,091	672.4	15,273	10,182	3,616.8	42.33	0.0017
2 Mbytes	5,066	673.9	15,198	10,132	3,596.0	42.33	0.0066
4 Mbytes	5,016	665.4	15,048	10,132	3,557.3	42.33	0.0164

표 5: 임의 접근, $U = 0.8$일때, NV cache 크기에 따른 디스크 쓰기 작업 감소 효과

NV Cache Size	Real Workload		Cleaning			Total	
	Num of Seg. W.	Time (sec)	Num of Seg. R.	Num of Seg. W.	Time (sec)	Mean Lifetime	Hit Ratio
1 Mbytes	4,792	578.6	6,234	1,449	1,082.0	97.62	0.0604
2 Mbytes	3,961	634.2	5,103	1,147	878.6	98.57	0.2232
4 Mbytes	2,486	312.3	3,093	608	518.1	102.07	0.5125

표 6: 지역성이 있는 접근, $U = 0.4$일때, NV cache 크기에 따른 디스크 쓰기 작업 감소 효과

NV Cache Size	Real Workload		Cleaning			Total	
	Num of Seg. W.	Time (sec)	Num of Seg. R.	Num of Seg. W.	Time (sec)	Mean Lifetime	Hit Ratio
1 Mbytes	4,791	578.6	18,895	14,104	4,653.2	32.20	0.0605
2 Mbytes	3,962	634.2	15,538	11,576	3,824.0	32.38	0.2230
4 Mbytes	2,485	312.3	9,720	7,235	2,388.0	32.46	0.5125

표 7: 지역성이 있는 접근, $U = 0.8$일때, NV cache 크기에 따른 디스크 쓰기 작업 감소 효과

이러한 부분 세그먼트를 제거하기 위한 방법의 하나로 비휘발성 메모리를 사용하였다. 그 결과 지역성이 있는 경우 비휘발성 메모리의 크기가 커질수록 쓰기 적중율이 크게 향상되었으며, 쓰기 적중율이 각각 대략 51 %, 6 % 일 경우, 공간 정리 비용이 두배 가까이 차이나는 것을 알 수 있었다.

추후에는 공간정리 정책, 디스테이징 알고리듬 및 세그먼트 크기 등을 다양하게 바꾸어 적용하였때의 부분 세그먼트 영향이 어떻게 변화하는가를 살펴볼 계획이다. 또한 디스크 배열로의 확장도 고려하고 있다.

참고 문헌

[1] M. Baker, E. D. S. Asami, J. Ousterhout, and M. Seltzer, "Non-volatile memory for fast, reliable file systems," in *Proc. 15th Int'l. CASPLOS*, Oct 1992.

[2] M. Rosenblum and J Ousterhout, "The design and implementation of a log-structered file system," in *Proc. 13th ACM SOSP*, Feb. 1992.

[3] M. F. K. Wiebren de Jonge and W. C. Hsieh, "The logical disk: A new approach to improving file systems," in *Proc. 14th ACM SOSP*, pp. 15–28, 1993.

[4] M. Seltzer, K. Bostic, M. K. McKusick, and C. Staelin, "An implementation of a log-structured file system for unix," in *USENIX Conference Proceedings*, (San Diego, CA), Jan. 1993.