

트랙 기반 MEMS 저장 장치 스케줄링 알고리즘

이소윤^o, 반효경
이화여자대학교

sounie^o@ewhain.net, bahn@ewha.ac.kr

Track-wise MEMS based Storage Scheduling Algorithm

Soyoon Lee^o, Hyokyung Bahn
EWha Womans University

요 약

MEMS 저장 장치는 기존의 디스크 저장 장치와는 다른 물리적 특성을 가지고 있는 차세대 저장 장치이다. 때문에 MEMS 저장 장치의 물리적 특성을 고려하는 디스크 스케줄링과는 다른 관리 기법이 필요하다. 본 연구에서는 MEMS 저장 장치의 물리적 특성을 고려한 트랙 기반 스케줄링 방법에 대하여 제안하였다. 트랙 기반 스케줄링 기법은 스케줄링에서의 문제 상태 공간과 스케줄링을 위한 연산 횟수를 줄이고 상대적으로 정착시간에 많은 크기를 갖는 물리적 특성을 고려하여 정착시간의 추가 횟수를 줄임으로써 요청에 대한 빠른 서비스를 가능하게 해주고 요청에 대한 대기 시간을 고려함으로써 동시에 기아상태를 방지한다.

1. 서 론

MEMS 기반 저장장치는 기존의 디스크보다는 10배 이상의 빠른 접근속도를 보이면서 전력소모는 1/100배의 효율을 보이며 플래시 메모리보다는 단위 공간 당 비용이 저렴한 차세대 저장장치이다. 그러나 MEMS 기반 저장장치는 기존의 디스크 저장장치와는 다른 물리적 특성을 가지고 있고 때문에 MEMS 저장 장치의 특성을 고려한 데이터 배치기법, 요청들에 대한 스케줄링 방법 등의 관리 기법들이 새롭게 연구되어지고 있다. 본 연구에서는 MEMS 저장 장치의 특성을 고려한 효율적인 스케줄링 기법으로 트랙 기반 스케줄링 접근 방법에 대하여 제안한다.

2. 배경 및 기존 연구

2.1. MEMS 기반 저장 장치

MEMS (MicroElectroMechanical System) 기반 저장 장치는 그림 1 에서와 같이 크게 데이터를 기록할 수 있는 2차원 평면의 매체(sled)와 데이터를 읽을 수 있는 여러 개의 고정 헤드(probe tip)로 구성되어 있다. 매체

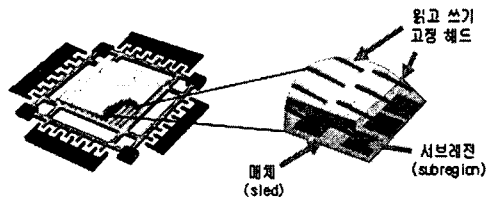


그림 1. MEMS 저장장치의 구조

는 여러 개의 서브 레전(subregion)으로 나누어져 있고 하나의 서브 레전에는 하나의 헤드가 존재하게 된다. 각 서브레전 마다의 헤드는 매체 위에 고정되어 있으며 매체의 x, y 축 움직임으로 헤드에 의해서 데이터를 읽을 수 있도록 되어 있다. 따라서, MEMS 저장장치에서 데이터의 위치는 $\langle x, y, tip \rangle$ 으로 표현된다[1].

일반적으로 디스크 저장장치는 데이터의 읽고 쓰기 연산을 위해 디스크 암(arm)이 해당 실린더의 위치를 찾기 위한 탐색시간(seek time)과 원하는 섹터에 도착하기 위한 회전시간(rotational time)이 필요하게 된다. 그러나 MEMS는 매체가 회전하지 않고 데이터를 읽기 위해 X, Y 축으로 동시에 움직여 원하는 데이터의 위치를 찾기 때문에 기존의 디스크 저장장치와는 다른 데이터 접근 비용을 정하게 된다. 그리고 여러 개의 헤드가 동시에 데이터를 접근하여 읽을 수 있는 특징을 가지고 있다.

2.2 기존 스케줄링 정책 및 성능척도

스케줄링 정책이란 다음으로 서비스를 받게 될 요청을 결정하는 방법에 있어서 어떠한 요소에 중점을 둘 것인가를 결정하는 것이다. 이렇게 정해진 정책에 따라 다음 요청을 결정하게 된다. 일반적으로 스케줄링 정책을 결정함에 있어서 고려해야 할 몇 가지 사항들이 있다. 현재 위치에서 다음 요청 후보지까지의 빠른 접근시간과 높은 전송량을 보장해주고, 기아 상태(starvation) 방지해 주면서 스케줄러는 효율적으로 구현되어야 한다는 것이다[2].

MEMS 기반 저장장치에 대한 스케줄링 정책도 이러한 고려 사항과 MEMS의 물리적 특징을 기초로 하여 연구

가 이루어져 왔다. 특히, 디스크 저장 장치의 스케줄링 방법으로 적용되었던 최소 탐색 시간 우선(SSTF: Shortest Seek Time First) 알고리즘, 최소 위치 시간 우선(SPTF: Shortest Positioning Time First) 알고리즘, 기아 상태를 방지하기 위해 SPTF 방법에 대기 시간 요소를 첨가시킨 Aging SPTF 알고리즘 등의 스케줄링 방법이 MEMS의 물리적 특징을 반영하여 적용되었다 [3]. 이것은 디스크에서 해당 실린더의 위치를 찾지 위한 디스크 암(arm)의 움직임과 디스크 헤드가 원하는 해당 섹터 위치로 도달하기까지의 회전을 각각 MEMS 저장장치에서의 X, Y 축으로의 움직임으로 대응시킴으로서 적용 가능하게 된다. 이 밖에도 기아 상태를 방지하기 위해 MEMS 저장장치를 여러 개의 지역으로 나누어 일정 순서대로 순회하도록 하고 지역 안에서는 빠른 접근 시간을 제공하기 위해서 SPTF 방법을 사용하는 지역 기반 SPTF(Zone-based SPTF) 알고리즘이 소개되었다 [4].

3. 트랙 기반 접근 방법(Track-wise approach)

SPTF 알고리즘은 x, y 축으로의 움직임을 고려했을 때, 현재 위치에서 가장 빠르게 접근 할 수 있는 곳의 요청을 먼저 서비스 하는 방법으로서 기존의 연구들에서 평균적으로 대기 중인 요청들에 대한 대기 시간을 줄이고 빠른 서비스를 제공해주는 효율적인 알고리즘으로 평가된다. 그러나 SPTF 알고리즘은 현재 위치로부터 다음 서비스 후보 요청지까지의 이동시간이 큰 경우 선택을 기피하게 되어 장시간 기다려도 서비스 요청으로 선택되어지지 않는 기아(starvation) 상태를 만들기 쉽다. 이것은 다음 스케줄링을 위한 요청들의 선택 기준이 이동시간이라는 지엽적인 기준에 의해서 선택되어지기 때문이다. 이동시간이라는 기준은 근시적인 관점에서는 대기 중인 요청들의 대기 시간을 고려하는 효율적인 방법으로 보이지만 항상 그 시점에서의 이동거리만을 고려하는 욕심쟁이(greedy) 방법으로 스케줄링 되어지기 때문에 앞에서 언급한 기아 상태 발생의 높은 우려를 가지고 있으며 최적 상태를 보장할 수 없다. 특히, 지엽적인 시각으로 스케줄링 되기 때문에 경우에 따라서는 요청들의 총 대기 시간이 비효율적인 결과를 가질 수도 있다.

우리가 제안하고 있는 트랙기반 알고리즘은 그림 2에서와 같이 다음 서비스 하게 될 요청을 선택함에 있어서의 기준은 이동시간과 각각의 요청에 대한 대기시간을 고려하게 된다. 여기서 이동시간은 트랙 별로 들어온 요청들에 대한 이동거리의 총합을 의미한다. 동일한 트랙에 들어온 요청들의 y축 위치를 기준으로 첫 번째 위치에서부터 마지막 요청의 지점까지의 모든 이동시간을 고려하여 이동시간에 대한 비용을 계산하게 된다. 이것은 SSTF, SPTF에서 제

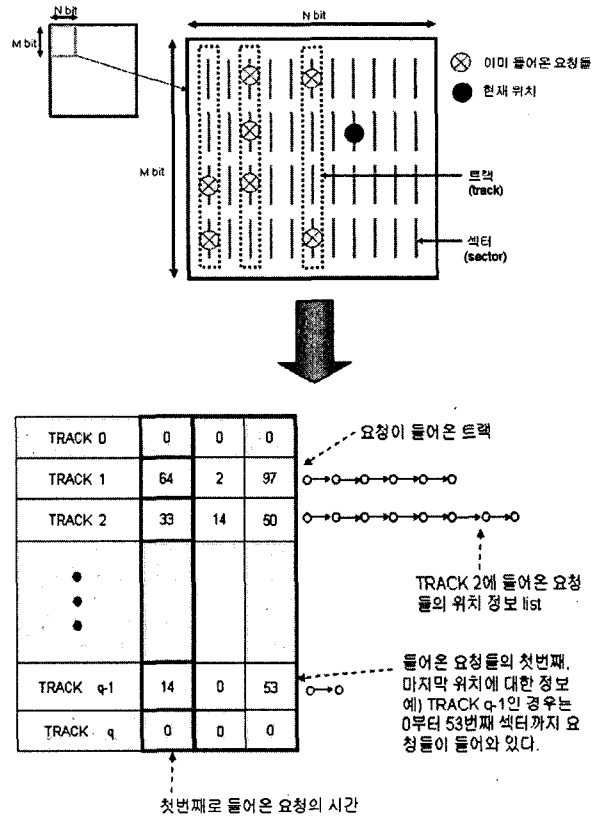


그림 2. 트랙 기반 접근 방법

$$COST(i) = T_{SEEK} - aW \quad (식 1)$$

$$T_{SEEK} = \max(t_x, t_y) + \sum y \text{ 축으로의 이동시간}$$

$$t_{sj} = \text{seek time} + \text{settling time}$$

$$t_y = \text{seek time} + \text{turnaround time}$$

W: 동일 트랙에서의 최장 대기 시간

T_{SEEK}: 현재 위치에서의 이동으로 인한 시간들의 합

t_x: 트랙의 첫 번째 위치로의 x축 이동시간.

t_y: 트랙의 첫 번째 위치로의 y축 이동시간

시한 이동 시간인 현재 위치에서 대기 요청들의 각각의 거리까지의 이동 시간과 구별된다. 또한 각각의 요청들에 대한 대기 시간은 기아상태를 줄이기 위한 요소로서 동일한 트랙에 들어온 요청 중 가장 첫 번째로 들어온 요청의 시간을 가지고 계산하게 된다. 비용 계산을 위한 식은 위의 식1과 같고 식1은 트랙 i위치에 들어온 요청들에 대한 비용을 의미한다.

일반적으로 스케줄링을 하기 위한 위치의 접근 단위는 섹터(sector)이다. 즉, 시작하는 위치의 섹터들을 기준으로 서비스하기 위해 드는 비용에 대하여 예측하게 된다. 알고리즘 별로 비용을 예측하는 기준은 달라지며 SPTF, SSTF와 같은 경우는 이동시간이 예측기준으로 제시된

다. 이러한 각각의 요청들에 대한 예측되는 비용을 기준으로 가장 작은 값을 지닌 요청이 다음에 서비스 하게 될 요청으로 선택되어지고 스케줄러는 이러한 비용 계산과 이에 대한 비교 연산을 하나의 요청에 대한 처리가 끝나고 다음 서비스 하게 될 요청을 선택해야하는 시점마다 반복하게 된다.

트랙을 기준으로 다음에 서비스 할 요청들에 대한 예상되는 비용을 계산하게 되는 트랙 기반 알고리즘을 사용하게 되면 비용의 계산과 비교 연산을 해야 하는 횟수가 줄어들게 된다. 일반적으로 비용 계산과 요청들에 대한 비용의 비교 연산은 섹터 기준인 경우 매 요청이 끝나는 시점마다 이뤄져야 한다. 그러나 트랙 기반 알고리즘에서는 서비스 하게 될 요청으로 선택되어지면 동일한 트랙에 들어온 여러 요청들이 Y축의 위치에 따라 비용의 재계산과 비교 연산 없이 순차적으로 처리되기 때문에 다음 서비스 하게 될 요청을 선택해야 하는 경우의 수가 줄어들게 된다.

스케줄링 하는 시점의 횟수가 감소하는 것뿐만 아니라 스케줄링을 위해 '비용 계산과 비교 연산을 하기 위한 문제 상태 공간도 줄어들게 된다. 즉, 기존의 방법들은 MEMS 저장 장치에 들어온 요청의 개수만큼 연산 대상이 확대되거나 트랙 기반으로 접근하게 되면 요청이 들어온 트랙의 개수만큼 비용계산과 비교 연산을 수행하면 된다. 최악의 경우에도 연산의 대상 범위인 문제 상태 공간의 크기는 트랙의 총 개수가 되고 이것은 항상 요청의 개수만큼 연산을 해야 하는 경우보다 작게 된다. 일반적으로 MEMS 저장 장치는 2500x2500 정도의 크기를 나타낸다[1]. 요청들 각각을 연산해야하는 경우는 문제 상태 공간이 6,250,000정도의 크기를 가지지만 트랙 기반인 경우는 2500 정도로 감소하게 된다.

MEMS 저장 장치는 한 섹터에서 시작하여 데이터를 읽기 시작할 때 Y축 방향으로 읽어나가기 시작하며 이것은 트랙(track)의 방향이다. 또한 MEMS 저장 장치에서 데이터를 읽어나가는 헤드는 X, Y축으로 동시에 움직이며 이동으로 발생하는 순수한 이동시간 뿐만 아니라 식 1에서와 같이 X, Y 축으로 각각 추가적인 시간이 소요되게 된다. X축으로는 원하는 위치에 착지하기 위해 드는 정착시간(settling time)이 Y축으로는 데이터를 읽고 쓰는 고정 헤드의 방향을 바꾸기 위한 회귀시간(turn-around time)이 더해진다[2]. 이것은 MEMS 저장 장치의 데이터를 담고 있는 매체(sled)가 스프링의 탄성력에 의해 움직이게 되는 물리적인 성질 때문이다. 일반적인 디스크 저장장치에도 정착시간은 존재하나 1-15 ms 정도의 디스크 접근 시간 중 0.5 ms 정도의 작은 비율을 차지한다.[1] 그러나 MEMS 저장 장치에서 정착시간은 0.2-0.8ms 정도의 장치 접근 시간 중 0.2ms 정도의 상대적으로 높은 비율을 차지함으로써 장치로의 접근 시간에서 중요한 요소로 간주되어지게 된다.[1] 즉, 정착시간

은 x축으로의 매체 움직임이 있을 경우에 추가되는 시간이기 때문에 최대한 x축으로의 이동을 자제하면서 스케줄링 하는 방법이 효율적일 수 있게 된다. 트랙 기반 접근 방법은 다음 서비스 하게 될 트랙으로 선택이 되면 그 트랙에 들어온 요청들을 부가적인 스케줄링 없이 처리해 주기 때문에 x축으로의 이동이 발생하는 것을 방지해주고 정착 시간의 발생을 감소시켜주게 된다. 이러한 효과는 SSTF, SPTF와 같이 현재 시점의 위치에서 가장 짧은 이동시간을 찾는 욕심쟁이 접근 방법과 비교하여 근시적으로 번잡한 매체의 이동으로 발생했던 추가적인 정착시간을 감소시킬 수 있게 된다.

MEMS 저장 장치의 데이터 배치 방법은 다양하게 제시되고 있으나 지역성을 감안한 배치방법(Columnar data layout)이 많이 사용되고 있으며 이 경우 배치 기법과 함께 고려되어 상승효과를 기대해 볼 수 있다.

4. 결 론

본 연구에서는 MEMS 기반 저장장치의 트랙 기반의 새로운 스케줄링 방법에 대하여 제안하였다. 트랙 기반 접근 방법은 스케줄링에 있어서의 문제 상태 공간과 스케줄링 시점 및 정착시간의 추가 횟수를 감소시키면서도 기아 상태를 방지하기 위해 최장 대기 시간을 감안하여 스케줄링 됨으로서 효율성을 보였다.

참고 문헌

- [1] Griffin, J., Schlosser, S., Ganger, G., Nagle, D., "Modeling and Performance of MEMS-Based Storage Devices," In Proceedings of ACM SIGMETRICS 2000, June 2000. Published as *Performance Evaluation Review*, 28(!):56-65. June 2000.
- [2] Silberschatz, Galvin, Gagne, "Operating system concepts sixth edition," Wiely, 2003.
- [3] Griffin, J., Schlosser, S., Ganger, G., Nagle, D., "Operating System Management of MEMS-based Storage Devices," In Proceedings 4th Symposium on Operating Systems Design & Implementation (OSDI 2000), October 2000.
- [4] Bo Hong, Scoot A. Brandt, Darrell D. E. Long, Ethan L. Miller, Karen A. Glocer, and Zachary N. J. Peterson, "Zone-based shortest positioning time first scheduling for MEMS-based storage devices," In Proceeding of the 11th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '03), pp. 104-113, Orlando, FL, 2003.