(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
G11B 19/14 (2006.01)
G11B 21/02 (2006.01)
G11B 20/10 (2006.01)

(45) 공고일자 2007년07월06일
(11) 등록번호 10-0736263
(24) 등록일자 2007년06월29일

(21) 출원번호 10-2006-0072728
(22) 출원일자 2006년08월01일
심사청구일자 2006년08월01일

(65) 공개번호
(43) 공개일자

(73) 특허권자 이화여자대학교 산학협력단
서울 서대문구 대현동 11-1 이화여자대학교내

(72) 발명자 반효경
서울시 관악구 봉천6동 1706 우성아파트 107-2001

문정민
충청남도 천안시 광덕면 내당리 좌무실 401-1번지

이소윤
서울시 강남구 압구정동 한양아파트 72동 512호

(74) 대리인 김석현

(56) 선행기술조사문헌
JP02152079 A
JP10199171 A

심사관 : 이강하

전체 청구항 수 : 총 5 항

(54) MEMS 기반 저장매체의 스케줄링 방법

(57) 요약

MEMS(Micro Elctro Mechanical System) 기반 저장매체의 스케줄링 방법이 개시된다.

본 발명은 수천 개의 고정된 헤드가 자신이 맡은 영역을 서비스하는 MEMS 기반 저장매체에 대한 요청을 스케줄링하는 방법에 관한 것으로, 상기 저장매체 상에 들어온 요청들을 병렬적 처리가 가능한 위치들로 분류하고, 이와 같은 분류를 토대로 각 위치에 대한 병렬적 처리 정도를 감안하여 요청들의 우선 순위를 계산하여 우선 순위를 결정하고 이로써 처리할 요청의 위치를 결정하여 스케줄링하는 것을 특징으로 한다.

따라서, 본 발명은 저장매체의 특성상 여러 헤드가 동시에 데이터를 읽고 쓸 수 있는 위치일수록 스케줄링의 우선 순위를 높임으로써 종래 스케줄링 방법에 따른 알고리즘에 비하여 월등한 평균 응답 시간을 구현할 수 있는 효과가 있다.

대표도

도 2

특허청구의 범위

청구항 1.

MEMS(Micro Elctro Mechanical System) 기반의 저장매체에 대한 요청들의 처리 순서를 결정하는 스케줄링 방법에 있어서,

(a) 상기 저장매체에 들어온 요청들을 병렬적 처리가 가능한 위치들로 분류하는 단계;

(b) 상기 각 위치에 대한 병렬 처리 정도를 포함하여 상기 요청들의 우선 순위를 계산하는 단계;

(c) 상기 저장매체에 들어온 모든 요청들에 대하여 상기 (a)단계 및 상기 (b)단계를 반복하고, 해당 위치에 대한 우선 순위를 결정하는 단계; 및

(d) 상기 (c)단계를 통해 결정된 우선 순위들을 비교한 결과에 따라 처리할 요청의 위치를 결정하는 단계

를 포함하는 스케줄링 방법.

청구항 2.

제1항에 있어서,

상기 (a)단계는,

서로 영역(region)은 다르지만 상기 영역 내의 상대적 위치는 동일하고 한 번의 탐색 시간으로 여러 헤드를 통해 동시에 요청을 처리할 수 있는 위치로 분류하는 것을 특징으로 하는 스케줄링 방법.

청구항 3.

제1항 또는 제2항에 있어서,

상기 (b)단계는,

상기 병렬적 처리 정도를 토대로 한 우선 순위를 계산하는 것을 특징으로 하는 스케줄링 방법.

청구항 4.

제3항에 있어서,

상기 (d)단계는,

상기 우선 순위의 크기를 비교하여 상기 우선 순위의 크기가 큰 순서대로 처리할 요청의 위치를 결정하는 것을 특징으로 하는 스케줄링 방법.

청구항 5.

제4항에 있어서,

상기 저장매체에 들어온 요청들은,

특정 시점을 기준으로 하여 상기 저장매체의 모든 영역의 각 위치에 대기 중인 모든 요청들인 것을 특징으로 하는 스케줄링 방법.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 MEMS 기반 저장매체의 스케줄링 방법에 관한 것으로서, 보다 세부적으로 입출력 처리를 수행하는 헤드가 각각 존재하는 복수의 영역으로 이루어진 MEMS 기반의 저장매체에 대한 요청들을 처리하기 위한 스케줄링 방법에 관한 것이다.

최근 멀티미디어 등 대용량 데이터의 폭발적 증가 및 휴대 기기 사용의 보편화에 따른 대용량 서버 스토리지와 모바일 스토리지 등 다양한 환경에서의 스토리지에 대한 용량 수요가 급격히 증가하는 추세이다.

이에 초소형 정밀기계 기술이라 일컬어 지는 MEMS(Micro Electro Mechanical System, 이하 'MEMS'라 기술함) 기반의 저장매체는 높은 대역폭과 저전력성, 고집적도, 저가 등의 특성으로 인해 모바일 스토리지와 대용량 서버 스토리지로 모두 사용될 수 있는 차세대 저장매체이다.

MEMS 기반의 저장매체는 하드디스크와 유사한 마그네틱 매체이지만 원판이 회전하고 헤드가 이동하는 하드디스크와 달리 사각형 구조의 매체가 스프링에 의해 *x*축 및 *y*축 방향으로 직접 움직이고 매체 위에 존재하는 수천 개의 고정된 헤드가 이를 읽고 쓰는 방식으로 동작한다.

도 1에서와 같이 매체는 수천 개의 영역(region)으로 나누어져 있고 각 영역마다 이를 전담하는 고정된 헤드가 하나씩 존재하여 서로 다른 영역에서는 동시에 데이터를 읽고 쓸 수 있다. 헤드가 영역 내의 특정 위치(x, y)에 있는 데이터를 읽고 쓰기 위해서는 해당 위치(x, y)로 이동하는 위치 탐색 시간(positioning time)이 필요하다.

그러나 헤드가 고정되어 있으므로 마그네틱 매체 자체가 스프링의 제어에 의해 읽고 쓰려는 특정 위치(x, y)를 헤드 쪽으로 이동시킨다.

이때, x축, y축 방향으로의 이동은 각각 독립적으로 동시에 진행된다.

이러한 물리적 구조 때문에 읽기 쓰기 동작에 있어 MEMS 기반 저장장치는 수천 개의 헤드에서 데이터를 병렬적으로 읽고 쓸 수 있다는 점과 데이터의 접근을 위해 *x*축과 *y*축 방향으로 동시에 매체를 이동시킨다는 점에서 하드디스크의 동작과 구별된다.

따라서 하드디스크의 경우 헤드의 1차원적 이동을 제어하는 거리 기반 스케줄링을 수행하지만, MEMS 기반 저장매체에서는 2차원 평면상에서 *x*축, *y*축 양방향의 이동을 제어해야 하므로 더욱 복잡한 스케줄링 문제가 발생한다.

종래의 MEMS 기반 저장매체의 스케줄링 알고리즘은 상기 MEMS 기반 저장매체의 물리적 특성들을 반영하지 않고 기존의 하드디스크에서 사용되던 스케줄링 알고리즘을 MEMS 기반 저장매체에 그대로 매핑한 알고리즘들이 대부분이며 병렬성을 고려한 스케줄링 알고리즘은 없었다.

이러한 기존의 기법들로는 제일 먼저 도착한 요청을 먼저 처리해주는 FCFS(First-Come First-Served), 헤드의 이동이 가장 적은 지점을 먼저 서비스하는 SSTF(Shortest Seek Time First), 헤드의 이동 시간(seek time)과 회전 지연 시간(rotational latency)을 함께 고려하는 SPTF(Shortest Positioning Time First) 등이 있다.

여기서, 하드디스크에서의 헤드 이동 시간(seek time)과 회전 지연 시간(rotational latency)을 각각 MEMS 기반 저장매체의 *x, y*축 방향 이동 시간으로 매핑할 경우 매우 유사한 결과를 얻을 수 있다.

그러나 이러한 하드디스크의 스케줄링 알고리즘을 매핑한 기법은 하드디스크에서는 볼 수 없었던 MEMS 기반 저장 장치만의 고유한 물리적 특성을 반영하지 못하고 있다.

발명이 이루고자 하는 기술적 과제

따라서, 본 발명이 이루고자 하는 기술적 과제는 수천 개의 헤드가 동시에 병렬적으로 데이터에 접근 가능하다는 MEMS 기반 저장매체의 물리적 특성을 고려한 스케줄링 기법을 제공하는 것이다.

발명의 구성

상기 기술한 바와 같은 과제를 이루기 위하여 본 발명의 특징에 따른 스케줄링 방법은,

MEMS(Micro Elctro Mechanical System) 기반의 저장매체에 대한 요청들의 처리 순서를 결정하는 스케줄링 방법에 있어서, (a) 상기 저장매체에 들어온 요청들을 병렬적 처리가 가능한 위치들로 분류하는 단계; (b) 상기 각 위치에 대한 병렬처리 정도를 포함하여 상기 요청들의 우선 순위를 계산하는 단계; (c) 상기 저장매체에 들어온 모든 요청들에 대하여 상기 (a)단계 및 상기 (b)단계를 반복하고, 해당 위치에 대한 우선 순위를 결정하는 단계; 및 (d) 상기 (c)단계를 통해 결정된 우선 순위들을 비교한 결과에 따라 처리할 요청의 위치를 결정하는 단계를 포함한다.

여기서, 상기 (a)단계는, 서로 영역(region)은 다르지만 상기 영역 내의 상대적 위치는 동일하고 한 번의 탐색 시간으로 여러 헤드를 통해 동시에 요청을 처리할 수 있는 위치로 분류하는 것을 뜻한다.

또한, 상기 (b)단계는, 상기 병렬적 처리 정도를 토대로 한 우선 순위를 계산하는 것을 뜻한다.

또한, 상기 (d)단계는, 상기 우선 순위의 크기를 비교하여 상기 우선 순위의 크기가 큰 순서대로 처리할 요청의 위치를 결정하는 것을 뜻한다.

또한, 상기 저장매체에 들어온 요청들은, 특정 시점을 기준으로 하여 상기 저장매체의 모든 영역의 각 위치에 대기 중인 모든 요청들인 것을 뜻한다.

아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다. 또한, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

이제, 본 발명의 실시예에 따른 MEMS 기반 저장매체의 스케줄링 방법에 대하여 도면을 참고로 하여 상세하게 설명한다.

도 2는 인입된 요청들을 MEMS 기반 저장매체에 평면적으로 나타낸 도면이며 동시에 병렬적으로 처리 가능한 요청들을 보여주는 도면이다.

도 2에 따르면, MEMS 기반 저장장치에 들어오는 읽고 쓰는 요청의 순서를 결정할 때, 서로 다른 영역(region)에 들어온 요청이지만 영역 내에서의 상대적인 위치가 동일하여 한 번의 헤드 접근으로 동시에 처리 가능한 요청들이 많은 경우 이 위치의 요청들에 대해 우선 순위를 높게 하여 먼저 처리해 준다.

즉 MEMS 기반 저장매체를 구성하는 다수의 헤드가 병렬적으로 입출력 처리를 수행할 수 있는 위치에 우선 순위를 높여 병렬적 처리가 가능한 위치가 먼저 처리되도록 하는 것이다.

이때, 요청들에 대한 병렬적 처리 가능 위치에 대한 우선순위 부여 방법은 다양하게 제시 가능하며 이러한 내용을 포함한 스케줄링 방법을 본 발명은 모두 포함하는 것이 바람직하다.

도 3은 본 발명의 제 1 실시 예에 따른 MEMS 기반 저장매체의 스케줄링 방법의 일련의 처리 과정을 보인 흐름도이다.

도 3에 따르면, 병렬적 처리가 가능한 위치의 요청들에게 높은 우선 순위를 부여하기 위해 각 위치에 대한 동시 처리 가능한 요청의 수 N(x, y)를 구하여 이 값이 높은 위치에 높은 우선 순위를 부여하는 일련의 처리 과정을 보인다.

즉, 저장매체에 대한 읽기, 쓰기 요청이 발생(S101)한 특정 위치(x, y)에 대해 상기 저장매체의 모든 영역에서의 요청의 수 즉 N(x, y)를 계산한다(S103).

여기서, 모든 영역에서의 요청의 수는 서로 영역(region)은 다르지만 영역내 상대적으로 동일한 위치에 있는 요청들의 개수를 의미한다. 이 위치의 요청들은 수천 개의 서로 다른 고정된 헤드에 의해 한 번의 탐색 시간으로 동시에 병렬적으로 처리되기 때문에 한꺼번에 큐에서 빠져나가 대기 중인 요청들의 수를 급격히 줄어들게 하는 효과를 나타낸다.

병렬적 처리 내용 외에 본 발명의 제 1실시예에서는 상기 특정 위치(x, y)에 대한 위치 탐색 시간을 고려한다(S105).

이때, 위치 탐색 시간은 각각 독립적으로 동시에 x축, y축 방향으로 진행되는 이동 시간을 의미한다. 이러한 위치 탐색 시간 $time_{position}(x, y)$는 아래 수학식 1과 같이 계산될 수 있다.

수학식 1

$$time_{position}(x, y) = \max(time_{seek\_x}, time_{seek\_y})$$

여기서, $time_{seek_x}$와 $time_{seek_y}$는 각각 x축과 y축 방향으로의 이동 시간을 의미한다.

상기 단계(S103)에서 계산한 요청의 수 및 상기 단계(S105)에서 계산한 위치 탐색 시간을 토대로 상기 특정 위치(x, y)에 대한 우선 순위(Priority(x, y))를 계산한다(S107).

이때, 우선 순위(Priority(x, y))는 아래 수학식 2와 같이 계산된다.

수학식 2

$$Priority(x, y) = N(x, y) / time_{position}(x, y)$$

저장매체에 대한 모든 요청들에 대하여 해당 위치에 대한 우선순위를 획득한다(S109).

여기서, 모든 요청들은 특정 시점을 기준으로 하여 상기 저장매체의 모든 영역의 각 위치에 대기 중인 요청들을 의미한다.

상기 단계(S109)를 통해 획득한 우선순위들을 비교(S111)한 결과에 따라 처리할 요청의 위치를 결정한다(S113). 즉 우선순위가 큰 값을 가지는 순서대로 요청을 처리한다.

도 4는 본 발명의 제 2 실시 예에 따른 MEMS 기반 저장매체의 스케줄링 방법의 일련의 처리 과정을 보인 흐름도이다.

도 4에 따르면, 동시에 처리 가능한 위치의 요청들에게 높은 우선순위를 부여해 주기 위해 직접적으로 동시 처리 가능한 요청의 개수를 이용하였던 도 3과는 달리 그 위치에서의 요청들의 대기 시간의 합을 이용하여 스케줄링한다.

동시에 처리 가능한 요청의 수가 많은 특정 위치(x, y)에서의 대기 중인 요청들의 대기 시간의 합($\sum W(x, y)$)은 대기 시간 자체는 큰 값을 가지지 않더라도 합해야 할 요청들이 많기 때문에 결과적으로 큰 값을 가진다. 따라서, 대기 중인 요청들의 대기 시간의 합을 고려하더라도 동시 처리 가능한 위치의 요청들에게 높은 우선순위를 부여할 수 있다.

또한, 도 4에 따르면, 요청들의 병렬적 처리 내용과 함께 요청 간의 형평성을 고려하여 기아 현상을 억제하기 위한 요소를 반영한다. 여기서, 기아 현상(starvation)은 현재 헤드의 위치에서 멀리 떨어진 곳이나 요청의 수가 적은 특정 위치는 스케줄링에서 계속적으로 소외되어 무한정 기다리게 되는 현상을 뜻한다. 도 4에 의하면 기다리는 시간이 길어질수록 대기 시간의 합(∑W(x, y))이 커지므로 우선 순위가 높아져 기아 현상을 억제하는 효과를 얻을 수 있다.

이와 같은 도 4에 의한 스케줄링 방법에 따르면, 먼저 저장매체의 특정 위치(x, y)에 대기 중인 요청들의 대기 시간의 합(∑W(x, y))을 계산한다(S301).

상기 특정 위치(x, y)에 대한 위치 탐색 시간($time_{position}(x, y)$)을 계산한다(S303).

상기 단계(S301)에서 산출한 대기 시간의 합 및 상기 단계(S303)에서 산출한 위치 탐색 시간을 토대로 상기 특정 위치에 대한 우선순위(Priority(x, y))를 계산한다(S305). 이때, 상기 우선순위(Priority(x, y))는 아래 수학식 3을 이용하여 계산한다.

수학식 3

$$Priority(x, y) = \sum W(x, y) / time_{position}(x, y)$$

여기서, $time_{position}(x, y)$는 특정 위치(x, y)로 이동하는데 걸리는 위치 탐색 시간을 의미하고, ∑W(x, y)는 특정 위치(x, y)에 있는 요청들의 대기 시간의 합을 의미한다.

상기 저장매체에 대한 모든 요청들에 대하여 상기 단계(S301, S303, S305)를 수행하여 해당 위치에 대한 우선순위를 획득한다(S307).

상기 단계(S307)를 통해 획득한 우선순위들을 비교한 결과에 따라 처리할 요청의 위치를 결정한다(S309, S311). 이때, 상기 우선순위가 큰 순서대로 처리한다.

이상 도 3 및 도 4를 통해 보인 스케줄링 방법에 따르면, MEMS 기반 저장매체에서 요청들에 대한 스케줄링을 함에 있어서 동시에 처리 가능한 위치에 높은 우선순위를 부여하여 그 위치의 요청들을 우선적으로 처리한다.

이와 같이 스케줄링을 하게 되면 서로 영역은 상이하지만 상대적으로 영역 내에서의 위치가 동일한 요청들은 수 천 개의 서로 다른 고정된 헤드에 의해 한번의 탐색시간으로 동시에 병렬적으로 처리되기 때문에 한꺼번에 큐에서 빠져나갈 수있다. 이로 인해 대기 중인 요청들의 수는 급격히 줄어들고 기존 알고리즘에 비해 요청들의 평균 응답시간이 짧아진다.

이제, 도 3과 도 4의 실시 예를 적용한 시뮬레이션 실험 결과를 설명하기로 한다.

먼저, 도 5는 시뮬레이션에 사용된 MEMS 기반 저장매체의 구조를 보인 도면이다.

도 5에 따르면, 저장매체는 6,400개의 영역이 있으며 각 영역은 x× y 평면상에서 2500× 2440 비트를 가진다. 각 영역은 2500× 27개의 섹터로 구성되며 섹터 하나는 8 바이트의 데이터가 인코딩된 80 비트로 구성된다.

그리고, 영역과 영역 사이에는 섹터 구분을 위한 10 비트의 부가 정보(servo information)가 저장된다. 또한, 512 바이트의 논리적 블록은 64개의 서로 다른 영역에서 상대적으로 같은 위치(x, y)의 8 바이트 섹터로 매핑하여 병렬적인 접근이 가능하도록 구성되었다.

또한, 인접한 논리적 블록들은 y축 방향을 따라 순차적으로 할당하여 부가적인 헤드 탐색 시간을 없애도록 하였다. 아래 표 1은 시뮬레이션에 사용된 MEMS 기반 저장매체의 구체적인 파라미터들을 나타낸다.

[표 1]

▪Number of regions	6400
▪Number of heads	6400
▪Maximum concurrent heads	1280
▪Device capacity	3.2 GB
▪Physical sector size	8 bytes
▪Servo overhead	10 bits per sector
▪Bits per region	2500 × 2440
▪Settling time	0.22 ms
▪Average turnaround time	0.07 ms
▪Spring factor	75%
▪Media bit cell size	40 × 40 nm
▪Sled acceleration	803.6 m/s^2
▪Sled access speed	28mm/s
▪Per head data rate	0.7 Mbit/s
▪Command processing overhead	0.2 ms/request
▪On-board cache memory	0MB

시뮬레이션의 성능 척도로는 평균 응답 시간을 사용한다. 이때, 시뮬레이션에 사용한 트레이스는 합성 트레이스(synthetic trace)와 실제 시스템의 추출 트레이스를 모두 사용한다.

합성 트레이스 경우, 다양한 평균 도착률을 지수 분포에 따라 생성하며 읽기와 쓰기 연산의 비율은 다른 연구에서 널리 사용한 67%와 33%으로 한다. 요청된 데이터의 크기는 평균이 4Kbyte인 지수 분포에 따르고 요청들의 배치는 전체 MEMS 기반 저장매체 전역에 걸쳐 균일하게 분포시킨다.

실제 시스템의 추출 트레이스는 HP 연구소의 HP-UX 운영체제를 사용하는 시분할 시스템에서의 디스크 입출력 데이터를 수집한 트레이스로서, 'Cello99 디스크 트레이스'를 사용한다.

시뮬레이션은 부하의 범위를 다양하게 실험하기 위해 다양하게 조절한 요청들의 도착 간격을 이용한다. 예를 들어 스케일링 요소(scaling factor)가 2인 경우 부하는 원래 트레이스에 비해 2배로 빈번한 요청이 이루어지는 것을 의미한다.

도 6과 도 7은 각각 합성 트레이스(synthetic trace)와 Cello99 트레이스를 사용한 경우 평균 도착률의 증가에 따른 요청의 응답 시간을 보여주는 도면으로서, 도 6은 합성 트레이스(synthetic trace)를 사용한 경우 평균 도착률의 증가에 따른 FCFS , SSTF , SPTF , ASPTF , P-SPTF , PA-SPTF 스케줄링 기법의 평균 응답 시간을 비교한 도면이고, 도 7은 Cello99 트레이스를 사용한 경우 스케일링 요소 증가에 따른 FCFS , SSTF , SPTF , ASPTF , P-SPTF , PA-SPTF 스케줄링 기법의 평균 응답 시간을 비교한 도면이다.

제 1 실시 적용 예인 도 3의 내용은 P-SPTF(Parallelism-aware SPTF, 이하 'P-SPTF'라 기술함)로, 제 2 실시 적용 예인 도 4의 내용은 PA-SPTF(Parallelism-aware with Aging extension SPTF, 이하 'PA-SPTF'라 기술함) 알고리즘으로 명명되었다.

이러한 적용 예들은 앞에서 설명한 병렬성을 고려하지 않은 기존의 스케줄링 방법인 FCFS(First-Come First-Served, 이하 'FCFS'라 기술함), SSTF(Shortest Seek Time First, 이하 'SSTF'라 기술함), SPTF(Shortest Positioning Time First, 이하 'SPTF'라 기술함), ASPTF(Aged Shortest Positioning Time First, 이하 'ASPTF'라 기술함) 스케줄링 기법과 성능을 비교하였다.

이 때, 도 6과 도 7에서 볼 수 있듯이 P-SPTF와 PA-SPTF는 요청의 평균 도착률이 증가함에 따라 FCFS, SSTF, SPTF, ASPTF보다 큰 성능 향상을 보임을 알 수 있다.

특히, 도 7에서 볼 수 있듯이, 'Cello99 트레이스'를 사용하는 경우에 스케일링 요소가 150 이상인 경우 P-SPTF와 PA-SPTF는 SPTF보다 39.2%에 이르는 성능 향상을 나타냄을 알 수 있다.

이는 영역 내의 상대적 위치가 동일한 요청들에 우선 순위를 높여 주는 동시에 헤드의 이동 시간을 줄였기 때문으로 평가된다.

이러한 실험 결과를 토대로 P-SPTF와 PA-SPTF는 FCFS, SSTF, SPTF, ASPTF에 비해 확장성이 매우 높아 멀티미디어 서버나 웹 서버처럼 대용량의 스트림 요청을 처리해야 하는 환경에서 사용할 경우 효율적일 것이라는 것을 알 수 있다.

이상에서 설명한 본 발명의 실시예는 장치 및 방법을 통해서만 구현이 되는 것은 아니며, 본 발명의 실시예의 구성에 대응하는 기능을 실현하는 프로그램 또는 그 프로그램이 기록된 기록 매체를 통해 구현될 수도 있으며, 이러한 구현은 앞서 설명한 실시예의 기재로부터 본 발명이 속하는 기술분야의 전문가라면 쉽게 구현할 수 있는 것이다.

그리고 본 발명의 권리범위는 상술한 실시예에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

발명의 효과

이와 같이 본 발명의 실시예에 의하면, 2차원 평면상의 헤드의 이동 시간과 동시 처리가 가능한 요청의 수를 고려하여 우선 순위를 결정하는 스케줄링을 통해 요청들에 대한 빠른 평균 응답 시간을 얻을 수 있다. 또한, 노화 요소를 추가하여 기아 현상을 극복하고 요청들의 대기 시간에 대한 형평성을 확보할 수 있다.

도면의 간단한 설명

도 1은 본 발명의 적용 분야인 일반적인 MEMS 기반의 저장매체의 물리적 구조를 보인 도면이다.

도 2는 인입된 요청들을 MEMS 기반 저장매체에 평면적으로 나타낸 도면이며 동시에 병렬적으로 처리 가능한 요청들을 보여주는 도면이다.

도 3은 본 발명의 제 1 실시 예에 따른 MEMS 기반 저장매체의 스케줄링 방법의 일련의 처리 과정을 보인 흐름도이다.

도 4는 본 발명의 제 2 실시 예에 따른 MEMS 기반 저장매체의 스케줄링 방법의 일련의 처리 과정을 보인 흐름도이다.

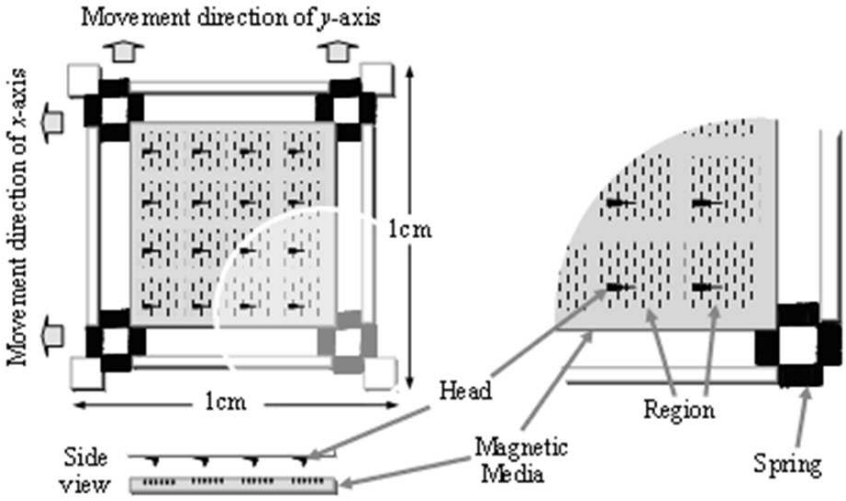
도 5는 시뮬레이션에 사용된 MEMS 기반 저장매체의 구조를 보인 도면이다.

도 6은 합성 트레이스(synthetic trace)를 사용한 경우 평균 도착률의 증가에 따른 FCFS , SSTF , SPTF , ASPTF , P-SPTF , PA-SPTF 스케줄링 기법의 평균 응답 시간을 비교한 도면이다.

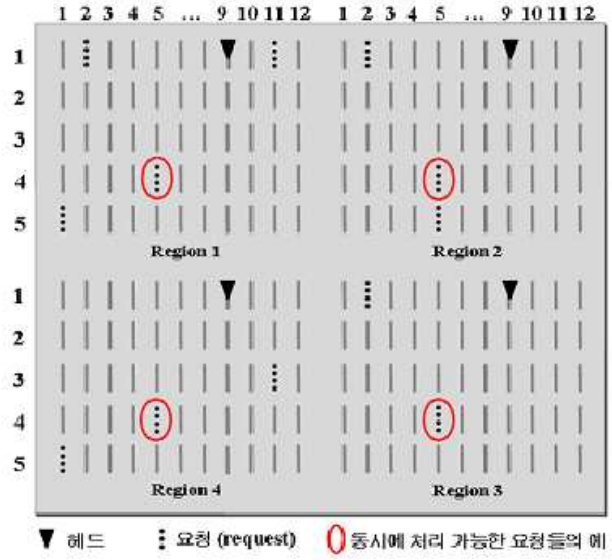
도 7은 Cello99 트레이스를 사용한 경우 스케일링 요소 증가에 따른 FCFS , SSTF , SPTF , ASPTF , P-SPTF , PA-SPTF 스케줄링 기법의 평균 응답 시간을 비교한 도면이다.

도면

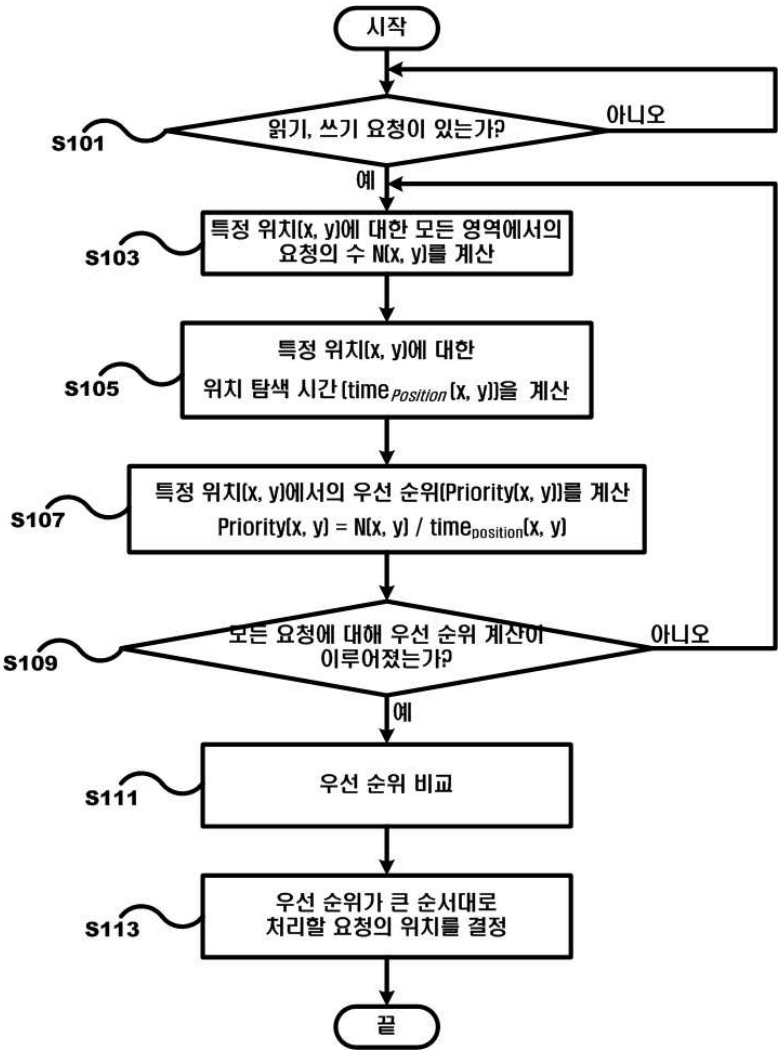
도면1


Movement direction of y-axis
Movement direction of x-axis
1cm
1cm
Head
Region
Magnetic Media
Spring
Side view

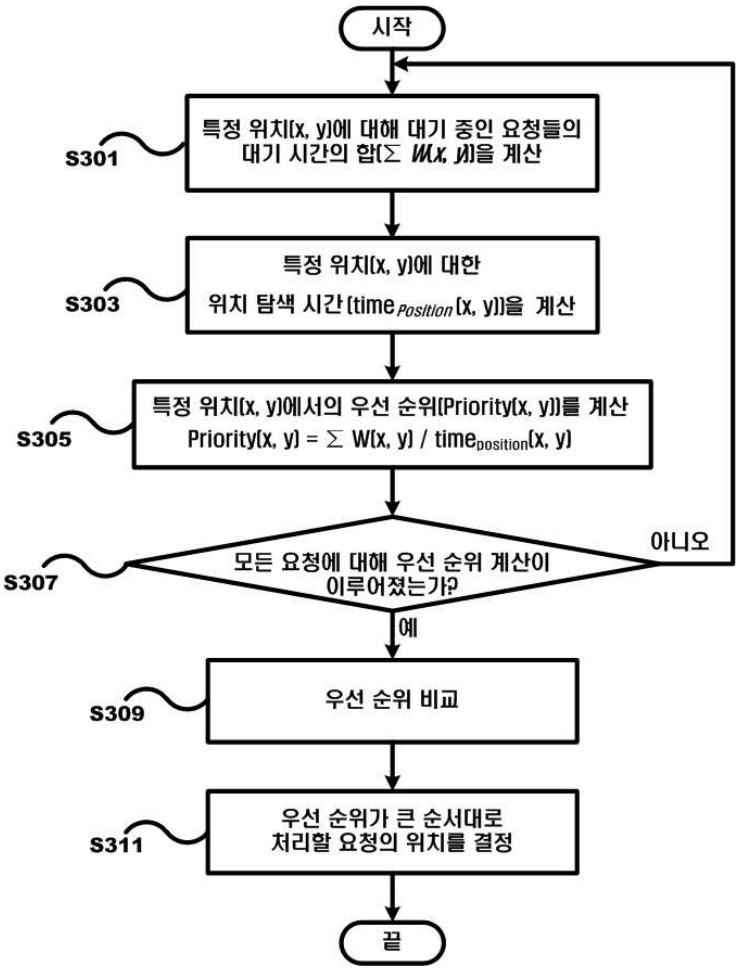
도면2


1 2 3 4 5 ... 9 10 11 12
1 2 3 4 5 ... 9 10 11 12
Region 1
Region 2
Region 4
Region 3
헤드
요청 (request)
동시에 처리 가능한 요청들의 예

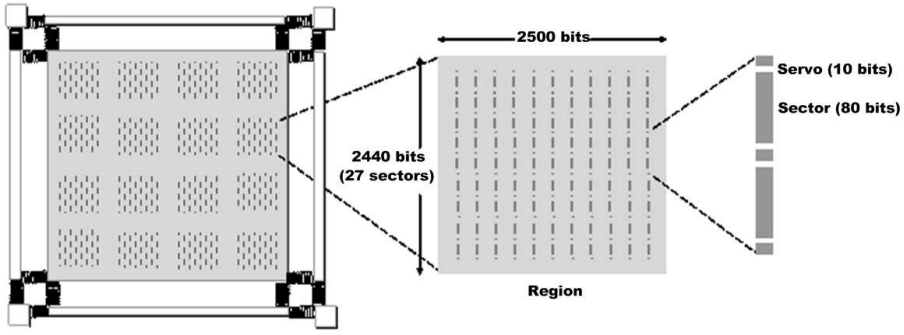
도면3


시작
S101
읽기, 쓰기 요청이 있는가?
아니오
예
S103
특정 위치(x, y)에 대한 모든 영역에서의 요청의 수 N(x, y)를 계산
S105
특정 위치(x, y)에 대한
위치 탐색 시간 (time Position (x, y))을 계산
S107
특정 위치(x, y)에서의 우선 순위(Priority(x, y))를 계산
Priority(x, y) = N(x, y) / time position(x, y)
S109
모든 요청에 대해 우선 순위 계산이 이루어졌는가?
아니오
예
S111
우선 순위 비교
S113
우선 순위가 큰 순서대로 처리할 요청의 위치를 결정
끝

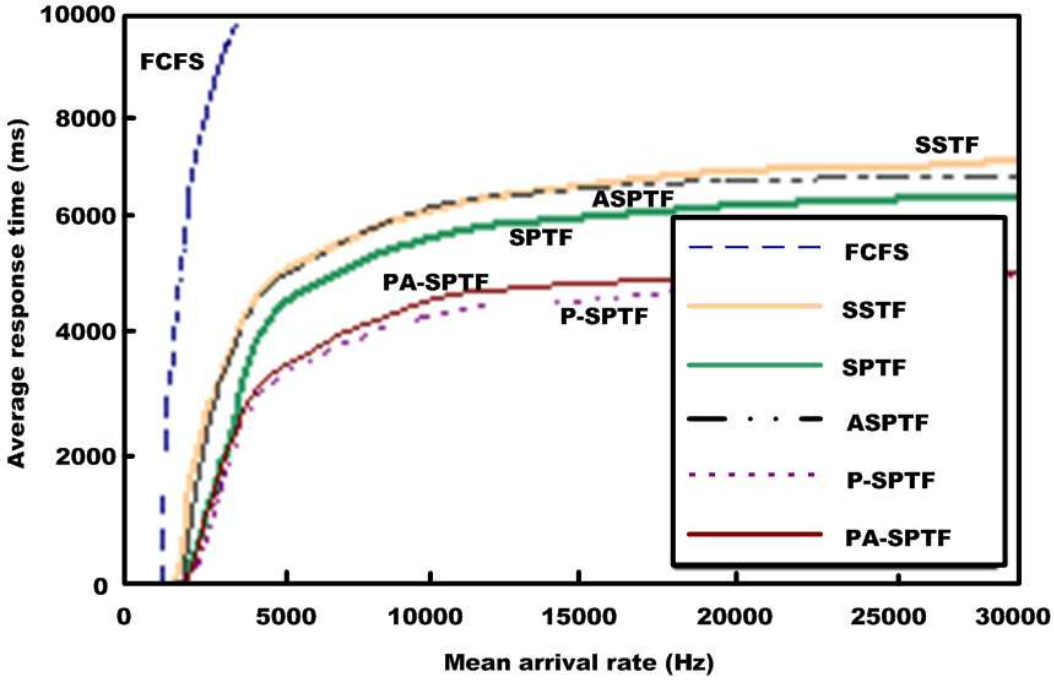
도면4


시작
S301
특정 위치(x, y)에 대해 대기 중인 요청들의 대기 시간의 합[Σ W(x, y)]을 계산
S303
특정 위치(x, y)에 대한 위치 탐색 시간 [time Position (x, y)]을 계산
S305
특정 위치(x, y)에서의 우선 순위(Priority(x, y))를 계산
Priority(x, y) = Σ W(x, y) / time position(x, y)
S307
모든 요청에 대해 우선 순위 계산이 이루어졌는가?
아니오
예
S309
우선 순위 비교
S311
우선 순위가 큰 순서대로 처리할 요청의 위치를 결정
끝

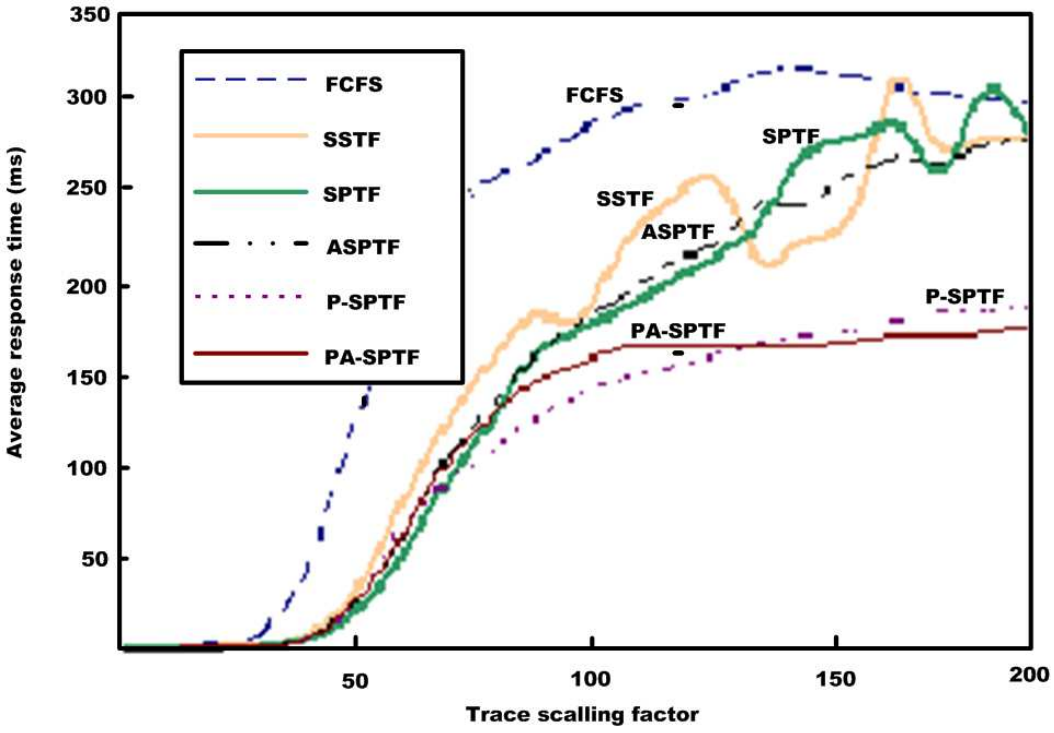
도면5


2500 bits
2440 bits
(27 sectors)
Servo (10 bits)
Sector (80 bits)
Region

도면6


10000
8000
6000
4000
2000
0
Average response time (ms)
0
5000
10000
15000
20000
25000
30000
Mean arrival rate (Hz)
FCFS
SSTF
ASPTF
SPTF
PA-SPTF
P-SPTF
FCFS
SSTF
SPTF
ASPTF
P-SPTF
PA-SPTF

도면7


350
300
250
200
150
100
50
Average response time (ms)
50
100
150
200
Trace scalling factor
FCFS
SSTF
SPTF
ASPTF
P-SPTF
PA-SPTF
FCFS
SPTF
SSTF
ASPTF
P-SPTF
PA-SPTF