



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2021년06월01일

(11) 등록번호 10-2259868

(24) 등록일자 2021년05월27일

(51) 국제특허분류(Int. Cl.)

G06F 13/16 (2006.01) G06F 12/08 (2016.01)

(52) CPC특허분류

G06F 13/1668 (2013.01)

G06F 12/0868 (2013.01)

(21) 출원번호 10-2015-0180958

(22) 출원일자 2015년12월17일

심사청구일자 2019년05월08일

(65) 공개번호 10-2017-0072600

(43) 공개일자 2017년06월27일

(56) 선행기술조사문헌

KR101521026 B1\*

KR1020150106132 A\*

\*는 심사관에 의하여 인용된 문헌

(73) 특허권자

에스케이텔레콤 주식회사

서울특별시 중구 을지로 65 (을지로2가)

연세대학교 산학협력단

서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)

(72) 발명자

노홍찬

서울특별시 중구 을지로 65 (을지로2가, SKT 타워)

박상현

서울특별시 양천구 오목로 300, 204동 3701호 (목동, 현대하이페리온2)

(뒷면에 계속)

(74) 대리인

특허법인 남앤남

전체 청구항 수 : 총 16 항

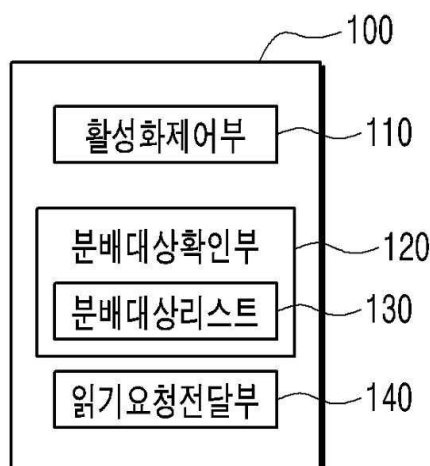
심사관 : 김세영

(54) 발명의 명칭 메모리제어장치 및 메모리제어장치의 동작 방법

### (57) 요약

본 발명은, 캐시 적중 데이터에 대한 읽기요청을 캐시메모리 뿐 아니라 하드디스크에서도 처리할 수 있도록 분배 전달함으로써, 전체 저장장치의 처리량(Throughput)을 극대화할 수 있는 메모리제어장치 및 메모리제어장치의 동작 방법을 개시하고 있다.

대표도 - 도2



(52) CPC특허분류

**G06F 12/0873** (2013.01)

(72) 발명자

**이종찬**

서울특별시 중구 을지로 65 (을지로2가, SKT 타워)

**박홍규**

서울특별시 중구 을지로 65 (을지로2가, SKT 타워)

---

**김재형**

인천광역시 부평구 안남로 272, 204동 2104호 (청천동, 금호아파트, 금호타운)

## 명세서

### 청구범위

#### 청구항 1

하드디스크 및 캐시메모리에 존재하는 캐시 적중 데이터에 대한 읽기요청이 확인되면, 캐시메모리의 상태에 따라서 분배 기능을 활성화시키는 활성화제어부;

상기 분배 기능이 활성화되면, 상기 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인하는 분배대상확인부; 및

상기 프로세스가 분배 대상인 것으로 확인되면 상기 읽기요청을 상기 하드디스크로 전달하여 상기 하드디스크에서 상기 읽기요청을 처리할 수 있게 하고, 상기 프로세스가 분배 대상이 아닌 것으로 확인되면 상기 읽기요청을 상기 캐시메모리로 전달하여 상기 캐시메모리에서 상기 읽기요청을 처리할 수 있게 하는 읽기요청전달부를 포함하는 것을 특징으로 하는 메모리제어장치.

#### 청구항 2

제 1 항에 있어서,

상기 활성화제어부는,

상기 캐시메모리에서 단위시간 동안 처리되는 요청의 평균 처리시간을 감시하여,

상기 평균 처리시간이 기 정의된 임계시간 이상이면, 상기 분배 기능을 활성화시키는 것을 특징으로 하는 메모리제어장치.

#### 청구항 3

제 2 항에 있어서,

상기 임계시간은,

상기 캐시메모리에서 단위시간 동안 처리되는 요청의 처리량에 따른 대역폭이 기 정해진 최대 대역폭에 근접한 때에 측정되는 최저 처리시간과 동일하거나 긴 시간인 것을 특징으로 하는 메모리제어장치.

#### 청구항 4

제 3 항에 있어서,

상기 활성화제어부는,

상기 분배 기능이 활성화된 상태에서, 상기 캐시메모리의 대역폭이 상기 최대 대역폭 보다 일정 크기 이상 작아지거나 또는 상기 하드디스크 및 캐시메모리의 총 대역폭이 상기 분배 기능이 활성화되기 직전 보다 작은 경우, 상기 분배 기능을 비활성화시키는 것을 특징으로 하는 메모리제어장치.

#### 청구항 5

제 2 항에 있어서,

상기 활성화제어부는,

상기 분배 기능을 활성화시키는 경우, 상기 평균 처리시간이 상기 임계시간 이상인 정도에 따라 분배 대상으로 설정 가능한 프로세스의 임계개수를 상이하게 정의하는 것을 특징으로 하는 메모리제어장치.

#### 청구항 6

제 1 항에 있어서,

상기 분배대상확인부는,

분배 대상으로 설정된 각 프로세스의 정보가 나열된 분배대상리스트에 기초하여, 상기 프로세스가 분배 대상이 되는지 여부를 확인하는 것을 특징으로 하는 메모리제어장치.

#### 청구항 7

제 6 항에 있어서,

상기 분배대상리스트에 나열된 프로세스의 정보는,

프로세스 구분자 및 프로세스에서 발생한 마지막 읽기요청의 요청시간을 포함하며,

상기 분배대상리스트는,

기 정의된 임계개수의 프로세스 정보가 나열된 상태에서 가장 오래된 요청시간을 포함하고 있는 프로세스의 정보가 테일(tail)에 위치하는 나열 구조를 가지며, 상기 분배대상확인부의 확인 결과를 반영하여 갱신되는 것을 특징으로 하는 메모리제어장치.

#### 청구항 8

제 7 항에 있어서,

상기 분배대상확인부는,

상기 프로세스의 정보가 상기 분배대상리스트에 존재하는 경우, 상기 프로세스를 분배 대상인 것으로 확인하며,

상기 분배대상리스트에서는,

상기 분배대상리스트에 존재하던 상기 프로세스의 정보가 삭제되고, 상기 분배대상리스트의 헤드부터 삭제된 상기 프로세스의 정보 직전까지 위치하는 각 프로세스의 정보가 이동된 후, 상기 읽기요청의 요청시간이 포함된 상기 프로세스의 정보가 헤드에 위치하도록 갱신되는 것을 특징으로 하는 메모리제어장치.

#### 청구항 9

제 7 항에 있어서,

상기 분배대상확인부는,

상기 프로세스의 정보가 상기 분배대상리스트에 존재하지 않는 경우, 분배 대상으로 설정된 프로세스의 개수가 상기 임계개수 미만이면 상기 프로세스를 분배 대상인 것으로 확인하며,

상기 분배대상리스트에서는,

상기 분배대상리스트의 헤드부터 위치하는 각 프로세스의 정보가 이동된 후, 상기 읽기요청의 요청시간이 포함된 상기 프로세스의 정보가 헤드에 위치하도록 갱신되는 것을 특징으로 하는 메모리제어장치.

#### 청구항 10

제 7 항에 있어서,

상기 분배대상확인부는,

상기 프로세스의 정보가 상기 분배대상리스트에 존재하지 않는 경우, 분배 대상으로 설정된 프로세스의 개수가 상기 임계개수 미만이면,

상기 분배대상리스트의 테일(tail)에 위치한 프로세스의 정보가 삭제 가능한지 확인하여, 삭제되는 경우에만 상기 프로세스를 분배 대상인 것으로 확인하는 것을 특징으로 하는 메모리제어장치.

#### 청구항 11

제 10 항에 있어서,

상기 분배대상리스트에서는,

상기 분배대상리스트의 테일에 위치한 프로세스 정보가 포함하고 있는 요청시간이 기 정의된 리스트유지시간을 경과한 경우 상기 테일에 위치한 프로세스 정보가 삭제되고,

상기 분배대상리스트의 헤드부터 위치하는 각 프로세스의 정보가 이동된 후, 상기 읽기요청의 요청시간이 포함된 상기 프로세스의 정보가 헤드에 위치하도록 갱신되는 것을 특징으로 하는 메모리제어장치.

#### 청구항 12

하드디스크 및 캐시메모리에 존재하는 캐시 적중 데이터에 대한 읽기요청이 확인되면, 캐시메모리의 상태에 따라서 분배 기능을 활성화시키는 활성화제어단계;

상기 분배 기능이 활성화되면, 상기 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인하는 분배대상확인단계; 및

상기 프로세스가 분배 대상인 것으로 확인되면 상기 읽기요청을 상기 하드디스크로 전달하여 상기 하드디스크에서 상기 읽기요청을 처리할 수 있게 하고, 상기 프로세스가 분배 대상이 아닌 것으로 확인되면 상기 읽기요청을 상기 캐시메모리로 전달하여 상기 캐시메모리에서 상기 읽기요청을 처리할 수 있게 하는 읽기요청전달단계를 포함하는 것을 특징으로 하는 메모리제어장치의 동작 방법.

#### 청구항 13

제 12 항에 있어서,

상기 분배대상확인단계는,

분배 대상으로 설정된 각 프로세스의 정보가 나열된 분배대상리스트에 기초하여, 상기 프로세스가 분배 대상이 되는지 여부를 확인하는 것을 특징으로 하는 메모리제어장치의 동작 방법.

#### 청구항 14

제 13 항에 있어서,

상기 분배대상리스트에 나열된 프로세스의 정보는,

프로세스 구분자 및 프로세스에서 발생한 마지막 읽기요청의 요청시간을 포함하며,

상기 분배대상리스트는,

기 정의된 임계개수의 프로세스 정보가 나열된 상태에서 가장 오래된 요청시간을 포함하고 있는 프로세스의 정보가 테일(tail)에 위치하는 나열 구조를 가지며, 상기 분배대상확인단계의 확인 결과를 반영하여 갱신되는 것을 특징으로 하는 메모리제어장치의 동작 방법.

#### 청구항 15

제 13 항에 있어서,

상기 분배대상확인단계는,

상기 프로세스의 정보가 상기 분배대상리스트에 존재하는 경우 상기 프로세스를 분배 대상인 것으로 확인하는 것을 특징으로 하는 메모리제어장치의 동작 방법.

#### 청구항 16

제 14 항에 있어서,

상기 분배대상확인단계는,

상기 프로세스의 정보가 상기 분배대상리스트에 존재하지 않는 경우,

분배 대상으로 설정된 프로세스의 개수가 상기 임계개수 미만이면 상기 프로세스를 분배 대상인 것으로 확인하고,

분배 대상으로 설정된 프로세스의 개수가 상기 임계개수 미만이면 상기 분배대상리스트의 테일(tail)에 위치한 프로세스의 정보가 삭제 가능한지 확인하여, 삭제되는 경우에만 상기 프로세스를 분배 대상인 것으로 확인하는 것을 특징으로 하는 메모리제어장치의 동작 방법.

## 발명의 설명

### 기술 분야

[0001] 본 발명은 메모리제어장치 및 메모리제어장치의 동작 방법에 관한 것으로, 더욱 상세하게는, 캐시 적중 데이터에 대한 읽기요청을 캐시메모리 뿐 아니라 하드디스크에서도 처리할 수 있도록 분배 전달함으로써, 전체 저장장치의 처리량(Throughput)을 극대화할 수 있는 메모리제어장치 및 메모리제어장치의 동작 방법에 관한 것이다.

### 배경 기술

[0002] 차세대 저장장치인 SSD(Solid State Disk)는 성능 측면에서 하드디스크 대비 우월한 장점이 있다. 하지만, SSD는 가격대비 용량 및 수명 측면에서 하드디스크 보다 비효율적인 단점이 있다.

[0003] 이와 같은 SSD의 하드디스크 대비 장/단점 때문에, 메모리 시스템에서는 SSD를 단일 저장장치로 사용하기 보다는 하드디스크의 캐시 디바이스로 사용하고 있다.

[0004] 이에, 메모리 시스템에서는, 비교적 작은 SSD를 캐시 디바이스(이하, SSD 기반 캐시메모리)로 사용해 낮은 가격으로 고효율적인 시스템을 구축하기 위해서, 자주 사용되는 데이터를 SSD 기반 캐시메모리에 적재하고 상대적으로 덜 빈번하게 사용되는 데이터를 하드디스크에 적재함으로써, 하드디스크를 단일 저장장치로 사용했을 때의 느린 성능과 SSD를 단일 저장장치로 사용했을 때의 높은 가격 및 제한된 수명을 완화하는 방식을 취하고 있다.

[0005] 따라서, SSD 기반 캐시메모리를 채택하는 메모리 시스템에서는 빈번하게 접근(사용)되는 데이터를 SSD 기반 캐시메모리에 적재하는 것이, 시스템 성능 면에서 유리할 것이다.

[0006] 한편, 메모리 시스템에서는, 캐시메모리의 특성상, 작은 크기의 데이터를 읽어오는 환경에서는 큰 문제가 없지만, 빅데이터 플랫폼과 같은 환경에서 대량의 데이터를 읽어오는 경우 비효율적인 부분이 존재한다.

[0007] 구체적으로 설명하면, 캐시메모리에서 데이터(캐시 적중 데이터)를 읽어오는 경우, 캐시메모리에서만 캐시 적중 데이터를 읽어오게 되고 그 동안 하드디스크는 동작하지 않게 되는데, 특히 캐시 적중 데이터가 대용량인 경우라면 하드디스크는 대용량의 캐시 적중 데이터가 캐시메모리에서 읽어지는 동안 계속 대기상태로 있게 되어, 하드디스크가 활용되지 못하기 때문에 저장장치의 처리량(Throughput) 측면에서 비효율적이라 할 것이다.

[0008] 이에, 본 발명에서는, 캐시 적중 데이터를 캐시메모리 뿐 아니라 하드디스크에서 읽어올 수 있도록 하여, 전체 저장장치의 처리량(Throughput)을 극대화할 수 있는 방안을 제안하고자 한다.

## 발명의 내용

### 해결하려는 과제

[0009] 본 발명은 상기한 사정을 감안하여 창출된 것으로서, 본 발명에서 도달하고자 하는 목적은 캐시 적중 데이터에 대한 읽기요청을 캐시메모리 뿐 아니라 하드디스크에서도 처리할 수 있도록 분배 전달함으로써, 전체 저장장치의 처리량(Throughput)을 극대화할 수 있는 메모리제어장치 및 메모리제어장치의 동작 방법을 제공하는데 있다.

### 과제의 해결 수단

[0010] 상기 목적을 달성하기 위한 본 발명의 제 1 관점에 따른 메모리제어장치는, 하드디스크 및 캐시메모리에 존재하는 캐시 적중 데이터에 대한 읽기요청이 확인되면, 캐시메모리의 상태에 따라서 분배 기능을 활성화시키는 활성화제어부; 상기 분배 기능이 활성화되면, 상기 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인하는 분배대상확인부; 및 상기 프로세스가 분배 대상인 것으로 확인되면 상기 읽기요청을 상기 하드디스크로 전달하여, 상기 캐시메모리 대신 상기 하드디스크에서 상기 읽기요청을 처리할 수 있게 하는 읽기요청전달부를 포함한다.

[0011] 바람직하게는, 상기 활성화제어부는, 상기 캐시메모리에서 단위시간 동안 처리되는 요청의 평균 처리시간을 감시하여, 상기 평균 처리시간이 기 정의된 임계시간 이상이면, 상기 분배 기능을 활성화시킬 수 있다.

[0012] 바람직하게는, 상기 임계시간은, 상기 캐시메모리에서 단위시간 동안 처리되는 요청의 처리량에 따른 대역폭이 기 정해진 최대 대역폭에 근접한 때에 측정되는 최저 처리시간과 동일하거나 긴 시간일 수 있다.

[0013] 바람직하게는, 상기 활성화제어부는, 상기 분배 기능이 활성화된 상태에서, 상기 캐시메모리의 대역폭이 상기

최대 대역폭 보다 일정 크기 이상 작아지거나 또는 상기 하드디스크 및 캐시메모리의 총 대역폭이 상기 분배 기능이 활성화되기 직전 보다 작은 경우, 상기 분배 기능을 비활성화시킬 수 있다.

- [0014] 바람직하게는, 상기 활성화제어부는, 상기 분배 기능을 활성화시키는 경우, 상기 평균 처리시간이 상기 임계시간 이상인 정도에 따라 분배 대상으로 설정 가능한 프로세스의 임계개수를 상이하게 정의할 수 있다.
- [0015] 바람직하게는, 상기 분배대상확인부는, 분배 대상으로 설정된 각 프로세스의 정보가 나열된 분배대상리스트에 기초하여, 상기 프로세스가 분배 대상이 되는지 여부를 확인할 수 있다.
- [0016] 바람직하게는, 상기 분배대상리스트에 나열된 프로세스의 정보는, 프로세스 구분자 및 프로세스에서 발생한 마지막 읽기요청의 요청시간을 포함하며, 상기 분배대상리스트는, 기 정의된 임계개수의 프로세스 정보가 나열된 상태에서 가장 오래된 요청시간을 포함하고 있는 프로세스의 정보가 테일(tail)에 위치하는 나열 구조를 가지며, 상기 분배대상확인부의 확인 결과를 반영하여 갱신될 수 있다.
- [0017] 바람직하게는, 상기 분배대상확인부는, 상기 프로세스의 정보가 상기 분배대상리스트에 존재하는 경우, 상기 프로세스를 분배 대상인 것으로 확인하며, 상기 분배대상리스트에서는, 상기 분배대상리스트에 존재하던 상기 프로세스의 정보가 삭제되고, 상기 분배대상리스트의 헤드부터 삭제된 상기 프로세스의 정보 직전까지 위치하는 각 프로세스의 정보가 이동된 후, 상기 읽기요청의 요청시간이 포함된 상기 프로세스의 정보가 헤드에 위치하도록 갱신될 수 있다.
- [0018] 바람직하게는, 상기 분배대상확인부는, 상기 프로세스의 정보가 상기 분배대상리스트에 존재하지 않는 경우, 분배 대상으로 설정된 프로세스의 개수가 상기 임계개수 미만이면 상기 프로세스를 분배 대상인 것으로 확인하며, 상기 분배대상리스트에서는, 상기 분배대상리스트의 헤드부터 위치하는 각 프로세스의 정보가 이동된 후, 상기 읽기요청의 요청시간이 포함된 상기 프로세스의 정보가 헤드에 위치하도록 갱신될 수 있다.
- [0019] 바람직하게는, 상기 분배대상확인부는, 상기 프로세스의 정보가 상기 분배대상리스트에 존재하지 않는 경우, 분배 대상으로 설정된 프로세스의 개수가 상기 임계개수 미만이면, 상기 분배대상리스트의 테일(tail)에 위치한 프로세스의 정보가 삭제 가능한지 확인하여, 삭제되는 경우에만 상기 프로세스를 분배 대상인 것으로 확인할 수 있다.
- [0020] 바람직하게는, 상기 분배대상리스트에서는, 상기 분배대상리스트의 테일에 위치한 프로세스 정보가 포함하고 있는 요청시간이 기 정의된 리스트유지시간을 경과한 경우 상기 테일에 위치한 프로세스 정보가 삭제되고, 상기 분배대상리스트의 헤드부터 위치하는 각 프로세스의 정보가 이동된 후, 상기 읽기요청의 요청시간이 포함된 상기 프로세스의 정보가 헤드에 위치하도록 갱신될 수 있다.
- [0021] 상기 목적을 달성하기 위한 본 발명의 제 2 관점에 따른 메모리제어장치의 동작 방법은, 하드디스크 및 캐시메모리에 존재하는 캐시 적중 데이터에 대한 읽기요청이 확인되면, 캐시메모리의 상태에 따라서 분배 기능을 활성화시키는 활성화제어단계; 상기 분배 기능이 활성화되면, 상기 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인하는 분배대상확인단계; 및 상기 프로세스가 분배 대상인 것으로 확인되면 상기 읽기요청을 상기 하드디스크로 전달하여, 상기 캐시메모리 대신 상기 하드디스크에서 상기 읽기요청을 처리할 수 있게 하는 읽기요청전달단계를 포함한다.
- [0022] 바람직하게는, 상기 분배대상확인단계는, 분배 대상으로 설정된 각 프로세스의 정보가 나열된 분배대상리스트에 기초하여, 상기 프로세스가 분배 대상이 되는지 여부를 확인할 수 있다.
- [0023] 바람직하게는, 상기 분배대상리스트에 나열된 프로세스의 정보는, 프로세스 구분자 및 프로세스에서 발생한 마지막 읽기요청의 요청시간을 포함하며, 상기 분배대상리스트는, 기 정의된 임계개수의 프로세스 정보가 나열된 상태에서 가장 오래된 요청시간을 포함하고 있는 프로세스의 정보가 테일(tail)에 위치하는 나열 구조를 가지며, 상기 분배대상확인단계의 확인 결과를 반영하여 갱신될 수 있다.
- [0024] 바람직하게는, 상기 분배대상확인단계는, 상기 프로세스의 정보가 상기 분배대상리스트에 존재하는 경우 상기 프로세스를 분배 대상인 것으로 확인할 수 있다.
- [0025] 바람직하게는, 상기 분배대상확인단계는, 상기 프로세스의 정보가 상기 분배대상리스트에 존재하지 않는 경우, 분배 대상으로 설정된 프로세스의 개수가 상기 임계개수 미만이면 상기 프로세스를 분배 대상인 것으로 확인하고, 분배 대상으로 설정된 프로세스의 개수가 상기 임계개수 미만이면 상기 분배대상리스트의 테일(tail)에 위치한 프로세스의 정보가 삭제 가능한지 확인하여, 삭제되는 경우에만 상기 프로세스를 분배 대상인 것으로

확인할 수 있다.

### 발명의 효과

[0026] 이에, 본 발명의 메모리제어장치 및 메모리제어장치의 동작 방법에 의하면, 캐시 적중 데이터에 대한 읽기요청을 캐시메모리 뿐 아니라 하드디스크에서도 처리할 수 있도록 분배 전달함으로써, 전체 저장장치의 처리량(Throughput)을 극대화할 수 있는 효과를 도출한다.

### 도면의 간단한 설명

[0027] 도 1은 본 발명의 바람직한 실시예에 따른 메모리제어장치가 포함된 메모리시스템의 구성을 간략하게 보여주는 구성도이다.

도 2는 본 발명의 바람직한 실시예에 따른 메모리제어장치를 구체적으로 보여주는 블록도이다.

도 3은 본 발명의 바람직한 실시예에 따른 분배대상리스트를 보여주는 예시이다.

도 4는 본 발명의 바람직한 실시예에 따른 메모리제어장치의 동작 방법을 나타내는 동작 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

[0028] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 대하여 설명한다.

[0029] 먼저, 도 1을 참조하여 본 발명의 바람직한 실시예에 따른 메모리제어장치가 포함된 메모리시스템의 구성을 설명하겠다.

[0030] 도 1에 도시된 바와 같이, 메모리시스템은, 저장장치로서의 하드디스크(20) 및 캐시메모리(10)와, 저장장치로의 메모리연산을 시도하는 파일시스템(200)과, 파일시스템(200) 및 저장장치 사이에서 저장장치를 제어하는 메모리제어장치(100)를 포함한다.

[0031] 이때, 본 발명의 메모리제어장치(100)는, 하드디스크(20)를 제어하는 하드디스크(20) 전용의 제어장치(미도시)와 연동이 가능한 캐시메모리(10) 전용의 제어장치일 수 있고, 캐시메모리(10) 및 하드디스크(20) 모두를 제어하는 제어장치일 수도 있다.

[0032] 이러한 본 발명의 메모리제어장치(100)는, 운영체제의 블록 계층 수준에서 동작하는 것이 바람직하다. 이는, 메모리제어장치(100)에 의해 제어되는 프로세스의 입/출력 형태가, 블록 계층에서 사용하는 자료구조에 따른다는 것을 의미하며, 그 형태는 입/출력의 크기(Sector 단위), 입/출력의 논리적 섹터 주소(Logical Sector Address), 입/출력의 형태(읽기/쓰기)를 포함할 수 있다.

[0033] 도 1에서 알 수 있듯이, 블록 계층 수준에서의 입/출력 형태, 즉 파일시스템(200) 구체적으로는 파일시스템(200) 내 프로세스에서 발생하는 읽기/쓰기요청의 구조체는, bio라는 자료구조에 따른다.

[0034] 따라서, 운영체제의 블록 계층에 위치하는 본 발명의 메모리제어장치(100)는, 파일시스템(200) 구체적으로는 파일시스템(200) 내 프로세스에서 발생하는 읽기요청(bio 구조체)을 제어할 수 있다.

[0035] 그리고, 본 발명의 메모리제어장치(100)가 적용되는 도 1의 메모리시스템에서, 캐시 디바이스로서 사용하는 캐시메모리(10)는 낸드 플래시 메모리인 것이 바람직하다.

[0036] 예를 들면, 캐시메모리(10)는, 낸드 플래시 메모리의 일종인 SSD(Solid State Drive)일 수 있다. 이하에서는 설명의 편의를 위해, 캐시메모리(10)와 SSD(10)를 혼용하여 사용하도록 하겠다.

[0037] 이와 같이, SSD(10)를 캐시 디바이스로서 사용하는 메모리시스템에서는, 비교적 작은 SSD(10)를 캐시 디바이스로 사용해 낮은 가격으로 고효율적인 시스템을 구축하기 위해서, 자주 사용되는 데이터를 SSD(10)에 적재하고 상대적으로 덜 빈번하게 사용되는 데이터를 하드디스크(20)에 적재함으로써, 하드디스크(20)를 단일 저장장치로 사용했을 때의 느린 성능과 SSD(10)를 단일 저장장치로 사용했을 때의 높은 가격 및 제한된 수명을 완화하는 방식을 취하고 있다.

[0038] 따라서, 메모리시스템에서는 빈번하게 접근(사용)되는 데이터를 SSD(10)에 적재하는 것이, 시스템 성능 면에서 유리할 것이다.

[0039] 한편, 메모리시스템에서는, 캐시메모리의 특성상, 작은 크기의 데이터를 읽어오는 환경에서는 큰 문제가



없지만, 빅데이터 플랫폼과 같은 환경에서 대량의 데이터를 읽어오는 경우 비효율적인 부분이 존재한다.

- [0040] 구체적으로 설명하면, SSD(10) 즉 캐시메모리(10)에서 데이터(캐시 적중 데이터)를 읽어오는 경우, 캐시메모리(10)에서만 캐시 적중 데이터를 읽어오게 되고 그 동안 하드디스크(20)는 동작하지 않게 되는데, 특히 캐시 적중 데이터가 대용량인 경우라면 하드디스크(20)는 대용량의 캐시 적중 데이터가 캐시메모리(10)에서 읽어드는 동안 계속 대기상태로 있게 되어, 하드디스크(20)가 활용되지 못하기 때문에 저장장치의 처리량(Throughput) 측면에서 비효율적이라 할 것이다.
- [0041] 이에, 본 발명에서는, 캐시 적중 데이터를 캐시메모리(10) 뿐 아니라 하드디스크(20)에서 읽어올 수 있도록 하여, 전체 저장장치의 처리량(Throughput)을 극대화할 수 있는 방안을 제안하고자 한다.
- [0042] 이하에서는, 도 2를 참조하여 본 발명의 바람직한 실시예에 따른 메모리제어장치를 구체적으로 설명하도록 한다.
- [0043] 본 발명에 따른 메모리제어장치(100)는, 하드디스크(20) 및 캐시메모리(10)에 존재하는 캐시 적중 데이터에 대한 읽기요청이 확인되면, 캐시메모리(10)의 상태에 따라서 분배 기능을 활성화시키는 활성화제어부(110)와, 분배 기능이 활성화되면, 상기 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인하는 분배대상확인부(120)와, 상기 프로세스가 분배 대상인 것으로 확인되면 상기 읽기요청을 하드디스크(20)로 전달하여, 캐시메모리(10) 대신 하드디스크(20)에서 상기 읽기요청을 처리할 수 있게 하는 읽기요청전달부(140)를 포함한다.
- [0044] 여기서 캐시메모리(10)는, 도 1을 참조한 설명에서 언급한 바와 같이, 낸드 플래시 메모리의 일종인 SSD(Solid State Drive)일 수 있다. 이하에서는 설명의 편의를 위해, 캐시메모리(10)와 SSD(10)를 혼용하여 사용하도록 하겠다.
- [0045] 활성화제어부(110)는, 하드디스크(20) 및 캐시메모리(10)에 존재하는 캐시 적중 데이터에 대한 읽기요청이 확인되면, 캐시메모리(10)의 상태에 따라서 분배 기능을 활성화시킨다.
- [0046] 보다 구체적으로 설명하면, 활성화제어부(110)는, 파일시스템(200) 구체적으로는 파일시스템(200) 내 프로세스에서 발생하는 읽기요청이 운영체제의 블록 계층에 위치한 본 발명의 메모리제어장치(100)로 수신(전달)되면, 금번 읽기요청이 캐시 적중 데이터에 대한 읽기요청인지 여부를 먼저 확인한다.
- [0047] 여기서, 캐시 적중 데이터에 대한 읽기요청인지 여부를 확인하는 것은, 기존의 확인 방식에 따르며, 이에 대한 구체적인 설명은 생략하도록 하겠다.
- [0048] 활성화제어부(110)에서 금번 읽기요청이 캐시 적중 데이터에 대한 읽기요청이 아니라고 확인되면, 본 발명의 메모리제어장치(100)는 금번 읽기요청을 하드디스크(20)로 전달하여, 하드디스크(20)에서 읽기요청을 처리할 수 있게 할 것이다.
- [0049] 활성화제어부(110)는, 금번 읽기요청이 캐시 적중 데이터에 대한 읽기요청으로 확인되면, 캐시메모리(10)의 상태에 따라서 분배 기능을 활성화시킨다.
- [0050] 즉, 본 발명에서는, 캐시메모리(10)의 상태가 혼잡상태에 놓이면, 분배 기능을 자동으로 활성화시키는 것이다.
- [0051] 이에 대해 보다 구체적으로 설명하면, 활성화제어부(110)는, 캐시메모리(10)에서 단위시간 동안 처리되는 요청의 평균 처리시간을 감시하여, 평균 처리시간이 기 정의된 임계시간 이상이면 분배 기능을 활성화시킬 수 있다.
- [0052] 활성화제어부(110)는, 캐시메모리(10)에서 단위시간(예: 1초) 동안 처리되는 요청(예: 읽기/쓰기요청)의 평균 처리시간을 감시한다.
- [0053] 즉, 활성화제어부(110)는, 캐시메모리(10)에서 단위시간(예: 1초) 동안 처리되는 각 요청(예: 읽기/쓰기요청)의 처리시간(await time)을 평균 연산한 평균시간을 감시하는 것이다.
- [0054] 이때, 처리시간(await time)이란, 요청(예: 읽기/쓰기요청)이 캐시메모리(10)에 전달되고, 해당 요청(예: 읽기/쓰기요청)이 캐시메모리(10)에서 최종적으로 수행 완료된 시점의 시간 차이를 의미한다.
- [0055] 이에, 활성화제어부(110)는, 단위시간(예: 1초) 단위로 감시한 평균 처리시간이 임계시간(await 임계값) 이상이면 분배 기능을 활성화시킬 수 있다.
- [0056] 여기서, 임계시간(await 임계값)은, 캐시메모리(10)에서 단위시간(예: 1초) 동안 처리되는 요청(예: 읽기/쓰기요청)의 처리량에 따른 대역폭이 기 정해진 최대 대역폭에 근접한 때에 측정되는 최저 처리시간(최저 await값)

과 동일하거나 긴 시간으로 정의되는 것이 바람직하다.

- [0057] 이를 위해, 본 발명에서는, 캐시메모리(10)에서 가장 최근의 단위시간(예: 1초) 동안 처리되는 요청(예: 읽기/쓰기요청)의 처리량에 따른 대역폭이 최대 대역폭에 근접한 때(예: 최대 대역폭  $\pm 5\%$ )에 최저 처리시간(최저 await값)을 찾아, 이 최저 처리시간(최저 await값)을 임계시간(await 임계값)으로 정의하거나, 또는 이 최저 처리시간(최저 await값) 보다 일정 크기 이상인 await값을 임계시간(await 임계값)으로 정의할 수 있다.
- [0058] 이에, 본 발명에서는, 전술과 같은 방식으로, 캐시메모리(10)에서 단위시간(예: 1초) 단위로 임계시간(await 임계값)을 정의할 수 있고, 이처럼 단위시간(예: 1초) 단위로 정의되는 임계시간(await 임계값)은 어느 정도의 시간이 경과하면서 자연스럽게 일정한 임계시간(await 임계값)으로 유지(정의)될 것이다.
- [0059] 따라서, 활성화제어부(110)는, 단위시간(예: 1초) 단위로 감시한 평균 처리시간이 전술과 같은 방식으로 정의된 임계시간(await 임계값) 이상이면, 캐시메모리(10)의 상태(처리량)가 최대 대역폭에 가까운 상태 즉 혼잡상태인 것으로 보고, 분배 기능을 자동으로 활성화시키는 것이다.
- [0060] 이때, 활성화제어부(110)는, 분배 기능을 활성화시키는 경우, 캐시메모리(10)에서 단위시간(예: 1초) 동안 처리되는 요청(예: 읽기/쓰기요청)의 평균 처리시간이 임계시간(await 임계값) 이상인 정도에 따라 분배 대상으로 설정 가능한 프로세스의 임계개수를 상이하게 정의할 수 있다.
- [0061] 예컨대, 활성화제어부(110)는, 평균 처리시간이 임계시간(await 임계값) 이상인 정도가 클수록 임계개수를 큰 개수로 정의하고 그 정도가 작을수록 임계개수를 작은 개수로 정의하도록 기 설정된 정책에 따라서, 임계개수를 상이하게 정의할 수 있다.
- [0062] 한편, 활성화제어부(110)는, 분배 기능이 활성화된 상태에서, 캐시메모리(10)의 대역폭 즉 캐시메모리(10)에서 가장 최근의 단위시간(예: 1초) 동안 처리되는 요청(예: 읽기/쓰기요청)의 처리량에 따른 대역폭이, 전술의 최대 대역폭 보다 일정 크기 이상 작아지면, 분배 기능을 자동으로 비활성화시킬 수 있다.
- [0063] 이는, 캐시메모리(10)의 대역폭이 최대 대역폭 보다 일정 크기 이상 작아지면, 캐시메모리(10)의 혼잡상태가 해소된 것으로 볼 수 있음에 기인한다.
- [0064] 또는, 활성화제어부(110)는, 분배 기능이 활성화된 상태에서, 하드디스크(20) 및 캐시메모리(10)의 총 대역폭이 분배 기능이 활성화되기 직전 보다 작은 경우, 분배 기능을 자동으로 비활성화시킬 수 있다.
- [0065] 이는, 분배 기능이 활성화된 상태에서, 하드디스크(20)에서 가장 최근의 단위시간(예: 1초) 동안 처리되는 요청(예: 읽기/쓰기요청)의 처리량에 따른 대역폭과 캐시메모리(10)에서 가장 최근의 단위시간(예: 1초) 동안 처리되는 요청(예: 읽기/쓰기요청)의 처리량에 따른 대역폭을 합한 총 대역폭이, 분배 기능이 활성화되기 직전의 총 대역폭 보다 작아지면, 분배 기능 활성화로 인해 오히려 전체 저장장치의 처리량(Throughput)이 감소한 것임에 기인한다.
- [0066] 이때, 활성화제어부(110)는, 분배 기능이 활성화된 상태에서, 하드디스크(20) 및 캐시메모리(10)의 총 대역폭이 분배 기능이 활성화되기 직전 보다 작은 경우, 앞서 정의된 임계시간(await 임계값)을 점진적으로 키움으로써, 분배 기능 활성화시키는데 불충분한 상황이 되도록 하는 방식으로 분배 기능 비활성화를 실현할 수도 있다.
- [0067] 분배대상확인부(120)는, 분배 기능이 활성화되면, 금번 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인한다.
- [0068] 보다 구체적으로, 분배대상확인부(120)는, 분배 대상으로 설정된 각 프로세스의 정보가 나열된 분배대상리스트에 기초하여, 금번 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인할 수 있다.
- [0069] 즉, 본 발명의 메모리제어장치(100)는, 분배 대상으로 설정된 각 프로세스의 정보가 나열된 구조의 분배대상리스트(130)를 관리하는 것이 바람직하다.
- [0070] 이에, 분배대상확인부(120)는, 분배대상리스트(130)에 기초하여, 금번 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인할 수 있다.
- [0071] 이하에서는, 분배대상리스트(130)에 대하여 구체적으로 설명하겠다.
- [0072] 분배대상리스트(130)는, 분배 대상으로 설정된 각 프로세스의 정보가 나열된 구조를 갖는다.
- [0073] 이때, 분배대상리스트(130)에 나열된 프로세스의 정보는, 프로세스 구분자 및 프로세스에서 발생한 마지막 읽기

요청의 요청시간을 포함하는 형태이다.

- [0074] 이에, 분배대상리스트(130)는, 가장 최근의 요청시간을 포함하고 있는 프로세스의 정보가 헤드(head)에 위치하고, 기 정의된 임계개수의 프로세스 정보가 나열된 상태에서 가장 오래된 요청시간을 포함하고 있는 프로세스의 정보가 테일(tail)에 위치하는 나열 구조를 가지는 것이 바람직하다.
- [0075] 그리고, 분배대상리스트(130)는, 분배대상확인부(120)의 확인 결과를 반영하여 갱신되는 것이 바람직하다.
- [0076] 도 3을 참조하여 보다 구체적으로 설명하면, 분배대상리스트(130)는, N개의 프로세스 정보를 나열 가능한 구조인 것이 바람직하다. 여기서, N개는, 전술한 활성화제어부(110)에 의해 정의된 임계개수일 것이다.
- [0077] 즉, 분배대상리스트(130)는, 헤드(head)부터 테일(tail)까지, 총 N개의 프로세스 정보를 나열할 수 있다.
- [0078] 이때, 전술한 바와 같이, 분배대상리스트(130)에 나열된 프로세스의 정보는, 프로세스 구분자(PID: Process ID)와, 프로세스에서 발생한 마지막 읽기요청의 요청시간을 포함하는 형태이다.
- [0079] 이에, 도 3을 참조하여 분배대상리스트(130)의 헤드에 위치한 프로세스의 정보를 언급하여 설명하면, 헤드에 위치한 프로세스1의 정보에는 프로세스1을 구분하기 위한 프로세스 구분자(PID1)과 프로세스1에서 발생한 마지막 읽기요청의 요청시간1이 포함된다.
- [0080] 이러한 분배대상리스트(130)는, 가장 최근의 요청시간을 포함하고 있는 프로세스의 정보가 헤드(head)에 위치하고, N개의 프로세스 정보가 나열된 상태에서 가장 오래된 요청시간을 포함하고 있는 프로세스의 정보가 테일(tail)에 위치하는 나열 구조를 가진다.
- [0081] 따라서, 도 3과 같이 N개의 프로세스 정보가 나열된 상태의 분배대상리스트(130)를 가정하면, 분배대상리스트(130)의 헤드에는 분배대상리스트(130)에 나열된 프로세스의 정보 중에서 마지막 읽기요청을 가장 최근에 발생한 프로세스1의 정보가 위치하고, 분배대상리스트(130)의 테일에는 분배대상리스트(130)에 나열된 프로세스의 정보 중에서 마지막 읽기요청을 가장 오래 전에 발생한 프로세스7의 정보가 위치하게 되는 것을 알 수 있다.
- [0082] 설명의 편의를 위해, 메모리제어장치(100)의 관리(갱신)에 의해 도 3과 같은 분배대상리스트(130) 상태를 가정하여 설명하겠다.
- [0083] 이하에서는, 금번 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인하는 과정을 구체적으로 설명하겠다.
- [0084] 분배대상확인부(120)는, 전술한 바와 같이, 캐시 적중 데이터에 대한 읽기요청이 확인되고 캐시메모리(10)의 상태가 혼잡상태인 경우, 금번 읽기요청을 발생한 프로세스의 정보가 분배대상리스트(130)에 존재하는지 확인한다.
- [0085] 이에, 분배대상확인부(120)는, 금번 읽기요청을 발생한 프로세스의 정보가 분배대상리스트(130)에 존재하면, 해당 프로세스를 분배 대상인 것으로 확인할 수 있다.
- [0086] 예를 들어, 금번 읽기요청을 발생한 프로세스가 프로세스3이라고 가정하면, 분배대상확인부(120)는, 금번 읽기요청을 발생한 프로세스3의 정보가 분배대상리스트(130)에 존재하는 것을 확인하고, 프로세스3을 분배 대상인 것으로 확인할 것이다.
- [0087] 이때, 분배대상확인부(120)의 확인 결과 즉 분배대상리스트(130)에 존재하는 프로세스3을 분배 대상으로 확인한 결과가, 분배대상리스트(130)에 반영되어 분배대상리스트(130)가 갱신될 것이다.
- [0088] 보다 구체적으로 분배대상리스트(130)에서는, 분배대상리스트(130)에 존재하던 프로세스3의 정보가 삭제되고, 분배대상리스트(130)의 헤드부터 삭제된 프로세스3의 정보 직전까지 위치하는 각 프로세스의 정보 즉 프로세스1,5의 정보가 이동(하나씩 뒤로 이동)된 후, 금번 읽기요청의 요청시간이 포함된 프로세스3의 정보가 헤드에 위치하도록 갱신될 것이다.
- [0089] 한편, 분배대상확인부(120)는, 금번 읽기요청을 발생한 프로세스의 정보가 분배대상리스트(130)에 존재하지 않는 경우, 분배 대상으로 설정된 프로세스의 개수가 임계개수 미만이면 해당 프로세스를 분배 대상인 것으로 확인할 수 있다.
- [0090] 예를 들어, 금번 읽기요청을 발생한 프로세스가 프로세스11이라고 가정하면, 분배대상확인부(120)는, 금번 읽기요청을 발생한 프로세스11의 정보가 분배대상리스트(130)에 존재하지 않는 것을 확인하고, 현재까지 분배 대상

으로 설정된 프로세스의 개수가 임계개수 미만인지 확인한다.

- [0091] 그리고, 분배대상확인부(120)는, 현재까지 분배 대상으로 설정된 프로세스의 개수가 임계개수 미만인 경우, 프로세스11을 분배 대상인 것으로 확인할 것이다.
- [0092] 여기서, 현재까지 분배 대상으로 설정된 프로세스의 개수가 임계개수(N) 미만인 경우라면, 분배대상리스트(130)에 나열된 프로세스의 정보 개수가 N개 보다 작은 경우이므로 분배대상리스트(130)에서 적어도 테일 부분은 비어 있을 것이다.
- [0093] 이때, 분배대상확인부(120)의 확인 결과 즉 분배대상리스트(130)에 존재하지 않는 프로세스11을 분배 대상으로 확인한 결과가, 분배대상리스트(130)에 반영되어 분배대상리스트(130)가 갱신될 것이다.
- [0094] 보다 구체적으로 분배대상리스트(130)에서는, 분배대상리스트(130)의 헤드부터 위치하는 각 프로세스의 정보 즉 프로세스1,5,3...의 정보가 이동(하나씩 뒤로 이동)된 후, 금번 읽기요청의 요청시간이 포함된 프로세스11의 정보가 헤드에 위치하도록 갱신될 것이다.
- [0095] 한편, 분배대상확인부(120)는, 금번 읽기요청을 발생한 프로세스의 정보가 분배대상리스트(130)에 존재하지 않는 경우, 분배 대상으로 설정된 프로세스의 개수가 임계개수(N) 미만이 아니면 분배대상리스트(130)의 테일에 위치한 프로세스의 정보가 삭제 가능한지 확인한다.
- [0096] 이에, 분배대상확인부(120)는, 분배대상리스트(130)의 테일에 위치한 프로세스의 정보가 삭제되는 경우에만, 프로세스11을 분배 대상인 것으로 확인할 것이다.
- [0097] 현재까지 분배 대상으로 설정된 프로세스의 개수가 임계개수(N) 미만이 아닌 경우라면, 분배대상리스트(130)는, 나열된 프로세스의 정보 개수가 N개로 테일까지 가득 차있는 풀(Full) 상태일 것이다.
- [0098] 이에, 분배대상리스트(130)에서는, 테일에 위치한 프로세스의 정보가 삭제 가능한지 확인이 시도됨에 따라, 분배대상리스트(130)의 테일에 위치한 프로세스7 정보가 포함하고 있는 요청시간7이 기 정의된 리스트유지시간을 경과한 경우 테일에 위치한 프로세스7의 정보가 삭제될 수 있다.
- [0099] 이는, 분배대상리스트(130)에서, 리스트유지시간 만큼 충분히 오래 전에 읽기요청을 발생한 분배 대상 프로세스의 정보를 삭제하고, 다른 새로운 프로세스를 분배 대상으로 설정하기 위함이다.
- [0100] 이에, 프로세스7 정보가 포함하고 있는 요청시간7이 리스트유지시간을 경과한 경우라면, 분배대상리스트(130)의 테일에서 프로세스7의 정보가 삭제될 것이다.
- [0101] 그리고, 분배대상확인부(120)는, 분배대상리스트(130)의 테일에 위치한 프로세스의 정보가 삭제되면, 프로세스11을 분배 대상인 것으로 확인할 것이다.
- [0102] 이때, 분배대상확인부(120)의 확인 결과 즉 분배대상리스트(130)에 존재하지 않는 프로세스11을 분배 대상으로 확인한 결과가, 분배대상리스트(130)에 반영되어 분배대상리스트(130)가 갱신될 것이다.
- [0103] 보다 구체적으로 분배대상리스트(130)에서는, 분배대상리스트(130)의 헤드부터 위치하는 각 프로세스의 정보 즉 프로세스1,5,3...의 정보가 이동(하나씩 뒤로 이동)된 후, 금번 읽기요청의 요청시간이 포함된 프로세스11의 정보가 헤드에 위치하도록 갱신될 것이다.
- [0104] 전술한 바와 같이, 분배대상리스트(130)는, 분배대상확인부(120)의 확인 결과를 반영하여 갱신되는 관리 방식을 통해, 어떤 프로세스가 분배 대상 프로세스(최대 N개 까지)인지, 그리고 분배 대상 프로세스가 발생한 마지막 읽기요청의 요청시간이 언제인지를 최신으로 관리하게 된다.
- [0105] 읽기요청전달부(140)는, 분배대상확인부(120)에서 금번 캐시 적중 데이터에 대한 읽기요청을 발생한 프로세스가 분배 대상인 것으로 확인되면, 금번 읽기요청을 하드디스크(20)로 전달한다.
- [0106] 즉, 읽기요청전달부(140)는, 분배 대상 프로세서가 발생한 읽기요청에 대해서는 하드디스크(20)로 전달하여, 캐시메모리(10) 대신 하드디스크(20)에서 읽기요청을 처리할 수 있게 하는 것이다.
- [0107] 물론, 읽기요청전달부(140)는, 분배대상확인부(120)에서 금번 캐시 적중 데이터에 대한 읽기요청을 발생한 프로세스가 분배 대상이 아닌 것으로 확인되면, 금번 읽기요청을 원래대로 캐시메모리(10)로 전달한다.
- [0108] 즉, 읽기요청전달부(140)는, 분배 대상 프로세서가 아닌 프로세스가 발생한 읽기요청에 대해서는 원래대로 캐시메모리(10)로 전달하여, 캐시메모리(10)에서 읽기요청을 처리할 수 있게 하는 것이다.



- [0109] 이상에서 설명한 바와 같이, 본 발명에 따르면, 캐시 적중 데이터에 읽기요청을 무조건 캐시메모리(10)로 전달하는 기준과 달리, 읽기요청을 캐시메모리(10) 뿐 아니라 하드디스크(20)에서도 처리할 수 있도록 분배 전달함으로써, 전체 저장장치의 처리량(Throughput)을 극대화할 수 있다.
- [0110] 더욱이, 본 발명에 따르면, 읽기요청을 캐시메모리(10) 및 하드디스크(20)로 분배함에 있어서, 어떤 프로세스가 분배 대상 프로세스(최대 N개 까지)인지 및 분배 대상 프로세스가 발생한 마지막 읽기요청의 요청시간이 언제인지를 지속적으로 갱신하여 최신으로 관리하는 분배대상리스트를 활용하는 프로세스 단위의 분배 방식을 제안함으로써, 읽기요청의 분배로 인한 시스템 성능 하락을 최소화하면서 효율적으로 분배할 수 있다.
- [0111] 이하에서는, 도 4를 참조하여 본 발명의 바람직한 실시예에 따른 메모리제어장치의 동작 방법을 설명하도록 하겠다. 설명의 편의를 위해서, 도 1 내지 도 3의 참조번호를 언급하여 설명하도록 하겠다.
- [0112] 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 파일시스템(200) 구체적으로는 파일시스템(200) 내 프로세스에서 발생하는 읽기요청이 운용체제의 블록 계층에 위치한 본 발명의 메모리제어장치(100)로 수신(전달)되면(S100), 금번 읽기요청이 캐시 적중 데이터에 대한 읽기요청인지 여부를 먼저 확인한다(S110).
- [0113] 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청이 캐시 적중 데이터에 대한 읽기요청이 아니라고 확인되면(S110 No), 금번 읽기요청을 하드디스크(20)로 전달하여(S160), 하드디스크(20)에서 읽기요청을 처리할 수 있게 할 것이다.
- [0114] 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청이 캐시 적중 데이터에 대한 읽기요청으로 확인되면(S110 Yes), 캐시메모리(10)의 상태에 따라서 분배 기능을 활성화시킨다(S120).
- [0115] 보다 구체적으로, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 캐시메모리(10)에서 단위시간(예: 1초) 동안 처리되는 요청(예: 읽기/쓰기요청)의 평균 처리시간을 감시하여, 평균 처리시간이 임계시간(await 임계값) 이상이면, 캐시메모리(10)의 상태(처리량)가 최대 대역폭에 가까운 상태 즉 혼잡상태인 것으로 보고, 분배 기능을 자동으로 활성화시키는 것이다(S120 Yes).
- [0116] 한편, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 단위시간(예: 1초) 단위로 감시한 평균 처리시간이 임계시간(await 임계값) 이상이 아니면, 캐시메모리(10)의 상태(처리량)가 혼잡상태가 아닌 것으로 보고, 분배 기능을 비활성화시킬 것이다(S120 No).
- [0117] 이에, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 분배 기능이 비활성화된 상태에서(S120 No), 금번 읽기요청을 캐시메모리(10)로 전달하여(S220), 캐시메모리(10)에서 읽기요청을 처리할 수 있게 할 것이다.
- [0118] 한편, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 분배 기능이 활성화된 상태에서(S120 Yes), 금번 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인한다.
- [0119] 이하에서는, 금번 읽기요청을 발생한 프로세스가 분배 대상이 되는지 여부를 확인하는 과정을 구체적으로 설명 하겠다.
- [0120] 먼저, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청을 발생한 프로세스의 정보가 분배대상리스트(130)에 존재하는지 확인한다(S130).
- [0121] 이에, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청을 발생한 프로세스의 정보가 분배대상리스트(130)에 존재하면(S130 Yes), 해당 프로세스를 분배 대상인 것으로 확인할 수 있다(S140).
- [0122] 예를 들어, 금번 읽기요청을 발생한 프로세스가 프로세스3이라고 가정하면, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청을 발생한 프로세스3의 정보가 분배대상리스트(130)에 존재하는 것을 확인하고, 프로세스3을 분배 대상인 것으로 확인할 것이다.
- [0123] 이때, S140단계의 확인 결과 즉 분배대상리스트(130)에 존재하는 프로세스3을 분배 대상으로 확인한 결과가, 분배대상리스트(130)에 반영되어 분배대상리스트(130)가 갱신될 것이다(S150).
- [0124] 보다 구체적으로 분배대상리스트(130)에서는, 분배대상리스트(130)에 존재하던 프로세스3의 정보가 삭제되고, 분배대상리스트(130)의 헤드부터 삭제된 프로세스3의 정보 직전까지 위치하는 각 프로세스의 정보 즉 프로세스 1,5의 정보가 이동(하나씩 뒤로 이동)된 후, 금번 읽기요청의 요청시간이 포함된 프로세스3의 정보가 헤드에 위치하도록 갱신될 것이다(S150).
- [0125] 그리고, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청을 하드디스크(20)로 전달하여

(S160), 하드디스크(20)에서 읽기요청을 처리할 수 있게 할 것이다.

- [0126] 한편, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청을 발생한 프로세스의 정보가 분배대상리스트(130)에 존재하지 않는 경우(S130 No), 분배 대상으로 설정된 프로세스의 개수가 기 정의된 임계개수(N) 미만인지 확인한다(S170).
- [0127] 예를 들어, 금번 읽기요청을 발생한 프로세스가 프로세스11이라고 가정하면, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청을 발생한 프로세스11의 정보가 분배대상리스트(130)에 존재하지 않는 것을 확인하고, 현재까지 분배 대상으로 설정된 프로세스의 개수가 임계개수(N) 미만인지 확인한다.
- [0128] 그리고, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 현재까지 분배 대상으로 설정된 프로세스의 개수가 임계개수(N) 미만인 경우(S170 Yes), 프로세스11을 분배 대상인 것으로 확인할 것이다(S180).
- [0129] 여기서, 현재까지 분배 대상으로 설정된 프로세스의 개수가 임계개수(N) 미만인 경우라면, 분배대상리스트(130)에 나열된 프로세스의 정보 개수가 N개 보다 작은 경우이므로 분배대상리스트(130)에서 적어도 테일 부분은 비어 있을 것이다.
- [0130] 이때, S180단계의 확인 결과 즉 분배대상리스트(130)에 존재하지 않는 프로세스11을 분배 대상으로 확인한 결과, 분배대상리스트(130)에 반영되어 분배대상리스트(130)가 갱신될 것이다(S190).
- [0131] 보다 구체적으로 분배대상리스트(130)에서는, 분배대상리스트(130)의 헤드부터 위치하는 각 프로세스의 정보 즉 프로세스1,5,3...의 정보가 이동(하나씩 뒤로 이동)된 후, 금번 읽기요청의 요청시간이 포함된 프로세스11의 정보가 헤드에 위치하도록 갱신될 것이다(S190).
- [0132] 그리고, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청을 하드디스크(20)로 전달하여(S160), 하드디스크(20)에서 읽기요청을 처리할 수 있게 할 것이다.
- [0133] 한편, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, S170단계에서 분배 대상으로 설정된 프로세스의 개수가 임계개수(N) 미만이 아니면(S170 No), 분배대상리스트(130)의 테일에 위치한 프로세스의 정보가 삭제 가능한지 확인한다.
- [0134] 현재까지 분배 대상으로 설정된 프로세스의 개수가 임계개수(N) 미만이 아닌 경우라면, 분배대상리스트(130)는, 나열된 프로세스의 정보 개수가 N개로 테일까지 가득 차있는 풀(Full) 상태일 것이다.
- [0135] 이에, 분배대상리스트(130)에서는, 테일에 위치한 프로세스의 정보가 삭제 가능한지 확인이 시도됨에 따라, 분배대상리스트(130)의 테일에 위치한 프로세스7 정보가 포함하고 있는 요청시간7이 기 정의된 리스트유지시간을 경과한 경우(S200 Yes) 테일에 위치한 프로세스7의 정보가 삭제될 수 있다(S210).
- [0136] 이는, 분배대상리스트(130)에서, 리스트유지시간 만큼 충분히 오래 전에 읽기요청을 발생한 분배 대상 프로세스의 정보를 삭제하고, 다른 새로운 프로세스를 분배 대상으로 설정하기 위함이다.
- [0137] 이에, 프로세스7 정보가 포함하고 있는 요청시간7이 리스트유지시간을 경과한 경우라면, 분배대상리스트(130)의 테일에서 프로세스7의 정보가 삭제될 것이다.
- [0138] 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 분배대상리스트(130)의 테일에 위치한 프로세스의 정보가 삭제되면(S210 ①), 프로세스11을 분배 대상인 것으로 확인할 것이다(S180).
- [0139] 이때, S180단계의 확인 결과 즉 분배대상리스트(130)에 존재하지 않는 프로세스11을 분배 대상으로 확인한 결과, 분배대상리스트(130)에 반영되어 분배대상리스트(130)가 갱신될 것이다(S190).
- [0140] 보다 구체적으로 분배대상리스트(130)에서는, 분배대상리스트(130)의 헤드부터 위치하는 각 프로세스의 정보 즉 프로세스1,5,3...의 정보가 이동(하나씩 뒤로 이동)된 후, 금번 읽기요청의 요청시간이 포함된 프로세스11의 정보가 헤드에 위치하도록 갱신될 것이다(S190).
- [0141] 그리고, 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 금번 읽기요청을 하드디스크(20)로 전달하여(S160), 하드디스크(20)에서 읽기요청을 처리할 수 있게 할 것이다.
- [0142] 본 발명에 따른 메모리제어장치(100)의 동작 방법은, 시스템이 종료되지 않는 한(S230 No), 읽기요청이 수신(전달)될 때마다 전술의 S100단계 이후를 수행함으로써(②), 읽기요청을 캐시메모리(10) 및 하드디스크(20)로 분배 전달할 것이다.

[0143] 이 상에서 설명한 바와 같이, 본 발명에 따른 메모리 제어 장치(100)의 동작 방법은, 캐시 적중 데이터에 읽기요청을 무조건 캐시메모리(10)로 전달하는 기존과 달리, 읽기요청을 캐시메모리(10) 뿐 아니라 하드디스크(20)에서도 처리할 수 있도록 분배 전달함으로써, 전체 저장장치의 처리량(Throughput)을 극대화할 수 있다.

[0144] 본 발명의 일실시예에 따른 메모리제어장치의 동작 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0145] 지금까지 본 발명을 바람직한 실시 예를 참조하여 상세히 설명하였지만, 본 발명이 상기한 실시 예에 한정되는 것은 아니며, 이하의 특허청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 또는 수정이 가능한 범위까지 본 발명의 기술적 사상이 미친다 할 것이다.

## 산업상 이용가능성

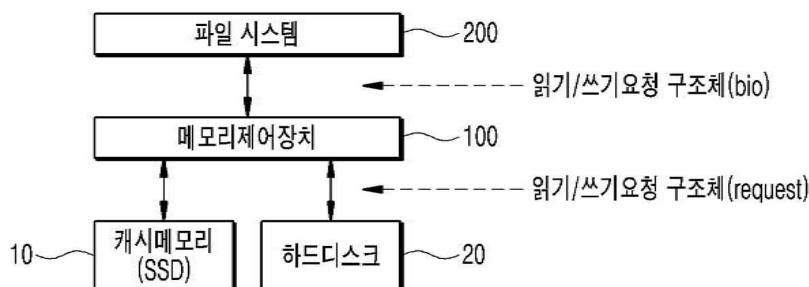
[0146] 본 발명에 따른 메모리장치 및 메모리장치의 동작 방법에 따르면, 캐시 적중 데이터에 대한 읽기요청을 캐시메모리 뿐 아니라 하드디스크에서도 처리할 수 있도록 분배 전달한다는 점에서, 기존 기술의 한계를 뛰어 넘음에 따라 관련 기술에 대한 이용만이 아닌 적용되는 장치의 시판 또는 영업의 가능성이 충분할 뿐만 아니라 현실적으로 명백하게 실시할 수 있는 정도이므로 산업상 이용가능성이 있는 발명이다.

## 부호의 설명

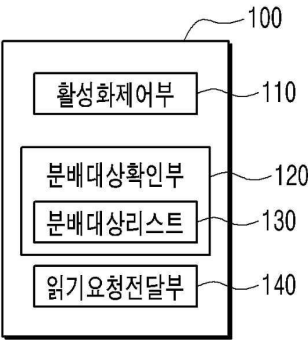
[0147]	10 : 캐시메모리	20 : 하드디스크
	100 : 메모리제어장치	
	110 : 활성화제어부	120 : 분배대상확인부
	130 : 분배대상리스트	140 : 읽기요청전달부
	200 : 파일시스템	

도면

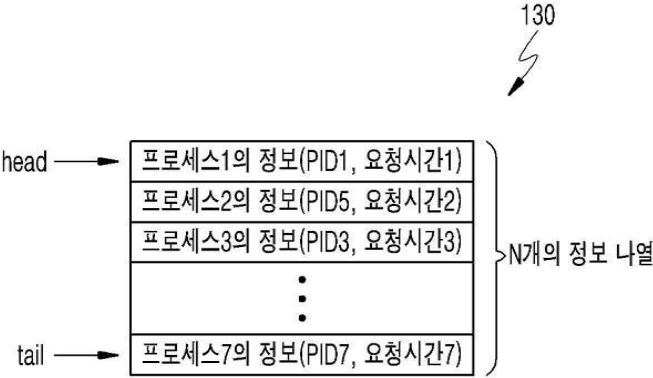
도면1



도면2

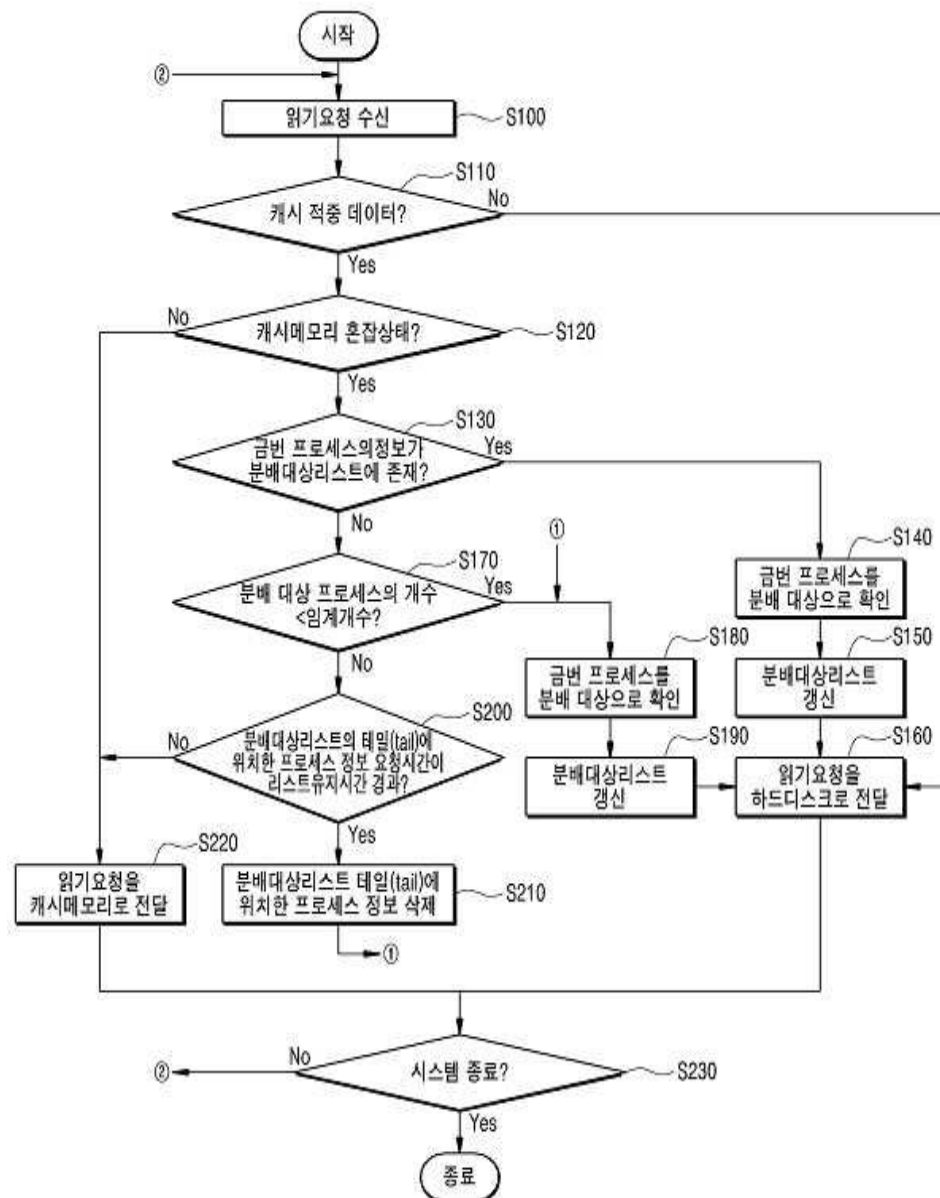


도면3





도면4



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 14

【변경전】

제 13 항에 있어서,

상기 분배대상리스트에 나열된 프로세스의 정보는,

프로세스 구분자 및 프로세스에서 발생한 마지막 읽기요청의 요청시간을 포함하며,

상기 분배대상리스트는,

기 정의된 임계개수의 프로세스 정보가 나열된 상태에서 가장 오래된 요청시간을 포함하고 있는 프로세스의 정보가 테일(tail)에 위치하는 나열 구조를 가지며, 상기 분배대상확단계의 확인 결과를 반영하여 갱신되는 것을 특징으로 하는 메모리제어장치의 동작 방법.

【변경후】

제 13 항에 있어서,

상기 분배대상리스트에 나열된 프로세스의 정보는,

프로세스 구분자 및 프로세스에서 발생한 마지막 읽기요청의 요청시간을 포함하며,

상기 분배대상리스트는,

기 정의된 임계개수의 프로세스 정보가 나열된 상태에서 가장 오래된 요청시간을 포함하고 있는 프로세스의 정보가 테일(tail)에 위치하는 나열 구조를 가지며, 상기 분배대상확인단계의 확인 결과를 반영하여 갱신되는 것을 특징으로 하는 메모리제어장치의 동작 방법.