



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2019년08월28일  
 (11) 등록번호 10-1982447  
 (24) 등록일자 2019년05월20일

(51) 국제특허분류(Int. Cl.)  
 G06F 9/455 (2018.01)

(52) CPC특허분류  
 G06F 9/45558 (2013.01)  
 G06F 2009/45579 (2019.08)

(21) 출원번호 10-2018-0084711

(22) 출원일자 2018년07월20일  
 심사청구일자 2018년07월20일

(56) 선행기술조사문헌  
 JP02196359 A\*  
 (뒷면에 계속)

(73) 특허권자

세종대학교산학협력단

서울특별시 광진구 능동로 209 (군자동, 세종대학교)

(72) 발명자

노재춘

서울특별시 도봉구 방학로15길 40, 107동 701호(방학동, 신동아아파트)

송광혁

경기도 남양주시 다산중앙로146번길 7, 2204동 203호(다산동, 다산 자연 앤 롯데캐슬)

(뒷면에 계속)

(74) 대리인

민영준

전체 청구항 수 : 총 3 항

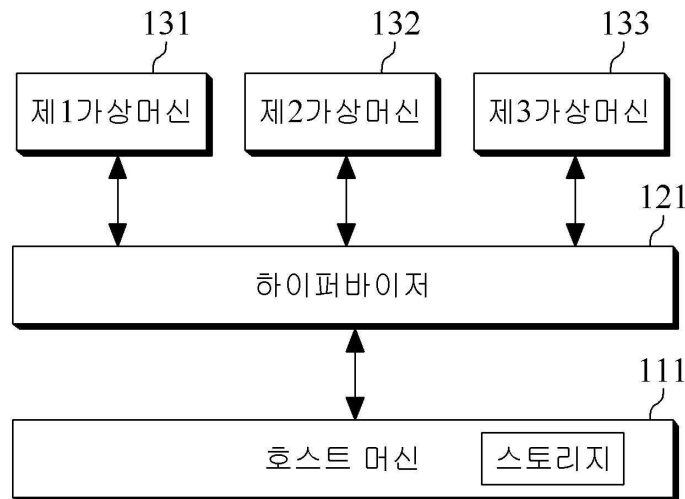
심사관 : 유진태

(54) 발명의 명칭 **가상 머신에 대한 입출력 처리 방법**

**(57) 요약**

가상 머신에 대한 입력 또는 출력 파일의 우선 순위에 따라 입출력 동작을 처리하는 방법이 개시된다. 개시된 가상 머신에 대한 입출력 처리 방법은 가상 머신에 대한 입력 또는 출력 파일의 메타데이터로부터, 상기 입력 또는 출력 파일에 대한 식별값을 추출하는 단계; 상기 식별값을 이용하여, 상기 입력 또는 출력 파일 각각에 대한 우선 순위 정보를 생성하는 단계; 및 상기 우선 순위 정보를 호스트 OS로 전달하는 단계를 포함한다.

**대표도** - 도1



- |   |  |
|---|--|
| <p>(52) CPC특허분류<br/>G06F 2209/484 (2013.01)</p> <p>(72) 발명자<br/><b>권세훈</b><br/>경기도 수원시 권선구 매탄로 25, 101동 207호(권선동, 삼천리권선2차아파트)</p> <p><b>최수미</b><br/>서울특별시 송파구 올림픽로 135, 233동 402호(잠실동, 리센츠)</p> | <p>(56) 선행기술조사문헌<br/>KR101579941 B1*<br/>JP63109564 A*<br/>JP2011165105 A*<br/>KR1020090063852 A<br/>JP5381713 B2<br/>KR1020160076148 A<br/>*는 심사관에 의하여 인용된 문헌</p> |
|---|--|

이 발명을 지원한 국가연구개발사업

과제고유번호	1345269682
부처명	교육부
연구관리전문기관	한국연구재단
연구사업명	개인기초연구
연구과제명	초고성능컴퓨팅플랫폼의 비용절감과 구성노드 경량화를위한 도커와 가상머신 기반의 하이브리드 가상화에 관한연구(2016R1D1A1B03930683)
기여율	1/2
주관기관	세종대학교
연구기간	2017.11.01 ~ 2018.08.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711070409
부처명	과학기술정보통신부
연구관리전문기관	정보통신기술진흥센터
연구사업명	대학ICT연구센터육성지원사업
연구과제명	모바일 플랫폼 기반 엔터테인먼트 VR 기술 연구
기여율	1/2
주관기관	세종대학교 산학협력단
연구기간	2018.01.01 ~ 2019.12.31

---

## 명세서

### 청구범위

#### 청구항 1

하이퍼바이저의 가상 머신에 대한 입출력 처리 방법에 있어서,  
 가상 머신에 대한 입력 또는 출력 파일의 메타데이터로부터, 상기 입력 또는 출력 파일에 대한 식별값을 추출하는 단계;  
 상기 식별값을 이용하여, 상기 입력 또는 출력 파일 각각에 대한 입출력 빈도값을 카운팅하는 단계; 및  
 상기 입출력 빈도값에 따라서, 우선 순위 정보를 생성하는 단계; 및  
 상기 우선 순위 정보를 호스트 OS로 전달하는 단계를 포함하며,  
 상기 입출력 빈도를 카운팅하는 단계는  
 미리 설정된 테이블의 전체 행의 개수를 상기 식별값으로 나눠 얻어진 나머지에 대응되는 상기 테이블의 행에, 상기 식별값을 대응시키는 단계; 및  
 상기 식별값에 대응된 행에, 상기 식별값에 대한 노드를 생성하는 단계를 포함하며,  
 상기 노드는, 상기 식별값의 개수에 따라서 카운팅되는 상기 입출력 빈도값을 포함하는  
 가상 머신에 대한 입출력 처리 방법.

#### 청구항 2

삭제

#### 청구항 3

삭제

#### 청구항 4

삭제

#### 청구항 5

제 1항에 있어서,  
 상기 우선 순위 정보를 생성하는 단계는  
 상기 입출력 빈도값에 대한 평균 및 표준편차를 이용하여, 상기 식별값에 대한 표준 점수를 생성하는 단계; 및  
 상기 표준 점수에 따라서, 상기 우선 순위 정보를 생성하는 단계를 포함하는 가상 머신에 대한 입출력 처리 방법.

#### 청구항 6

제 1항에 있어서,  
 상기 우선 순위 정보를 생성하는 단계는  
 상기 가상 머신에 대한 입력 또는 출력이 발생할 때마다, 상기 우선 순위 정보를 갱신하는

가상 머신에 대한 입출력 처리 방법.

**청구항 7**

삭제

**청구항 8**

삭제

**청구항 9**

삭제

**청구항 10**

삭제

**발명의 설명**

**기술 분야**

[0001] 본 발명은 가상 머신에 대한 입출력 처리 방법에 관한 것으로서, 더욱 상세하게는 가상 머신에 대한 입력 또는 출력 파일의 우선 순위에 따라 입출력 동작을 처리하는 방법에 관한 것이다.

**배경 기술**

[0003] 빅 데이터가 4차 산업시대의 중요 요소로 떠오르며 이를 처리하기 위한 방법으로 클라우드 컴퓨팅이 주목받고 있다. 클라우드 컴퓨팅의 핵심 요소인 분산 처리 기법에서는 빅 데이터를 병렬적으로 처리하기 위해 필수적이며 다수의 서버를 기반으로 하는 플랫폼이 요구된다. 이를 위해 한 대의 호스트 컴퓨터를 여러 대의 컴퓨터처럼 사용할 수 있는 가상화 기법이 이용된다.

[0004] 가상화는 컴퓨터의 CPU나 메모리, 디스크와 같은 자원의 물리적 구성을 은폐한 다음, 복수의 가상적인 하드웨어 환경을 만들어 각 이 CPU나 메모리를 갖추고 있는 것처럼 보이게 할 수 있는 소프트웨어 기술이다. 가상화를 통해 자원이 논리적으로 분배되기 때문에 시스템의 자원 가용성을 높일 수 있고 필요에 따라 자원을 추가하거나 감소시키는 것이 가능하여 확장이 용이하다.

[0005] 가상화 기술에서 하이퍼바이저(hypervisor)를 통해 하나의 호스트 머신에서 복수의 운영체제를 동작시킬 수 있다. 하이퍼바이저는 베어메탈(bare-metal) 방식과 호스티드(hosted) 방식으로 구분된다.

[0006] 가상화 시스템에서는 호스트 머신의 물리적인 스토리지 자원이 가상 머신 각각에 논리적으로 할당되고, 가상 머신 각각에 대한 파일 입출력(I/O, Input/Output)이 수행된다. 가상화 환경에서 파일 입출력이 많은 자원을 소모하기 때문에, 가상화 시스템의 성능을 높이기 위해서는 가상 머신에 대한 입출력 처리가 최적화되는 것이 중요하다.

[0007] 관련 선행문헌으로 특허 문헌인 대한민국 등록특허 제10-1790728호가 있다.

**발명의 내용**

**해결하려는 과제**

[0009] 본 발명은 가상 머신에 대한 입출력 처리 성능을 향상시킬 수 있는 방법을 제공하기 위한 것이다.

**과제의 해결 수단**

[0011] 상기한 목적을 달성하기 위한 본 발명의 일 실시예에 따르면, 가상 머신에 대한 입력 또는 출력 파일의 메타데이터로부터, 상기 입력 또는 출력 파일에 대한 식별값을 추출하는 단계; 상기 식별값을 이용하여, 상기 입력 또는 출력 파일 각각에 대한 우선 순위 정보를 생성하는 단계; 및 상기 우선 순위 정보를 호스트 OS로 전달하는

단계를 포함하는 가상 머신에 대한 입출력 처리 방법이 제공된다.

[0012] 또한 상기한 목적을 달성하기 위한 본 발명의 다른 실시예에 따르면, 가상 머신에 대한 입력 또는 출력 파일의 우선 순위 정보를 제공받는 단계; 상기 가상 머신에 대한 파일 입출력 요청이 발생한 경우, 상기 요청된 파일의 우선 순위 정보에 따라서, 상기 입출력 요청을 상기 우선 순위에 대응되는 서로 다른 큐에 저장하는 단계; 및 상기 우선 순위에 따라서, 상기 큐에 저장된 입출력 요청을 처리하는 단계를 포함하는 가상 머신에 대한 입출력 처리 방법이 제공된다.

**발명의 효과**

[0014] 본 발명에 따르면, 우선 순위가 높은 파일의 입출력을 우선적으로 처리함으로써, 가상 머신에 대한 입출력 성능을 향상시킬 수 있다.

[0015] 또한, 본 발명에 따르면, 사용자에게 의해 자주 사용되는 파일의 입출력이 상대적으로 빨리 처리됨으로써, 사용자의 체감 속도도 향상될 수 있다.

**도면의 간단한 설명**

[0017] 도 1은 본 발명의 일실시예에 따른 가상화 시스템을 설명하기 위한 도면이다.

도 2는 본 발명의 일실시예에 따른 가상 머신에 대한 입출력 처리 방법을 설명하기 위한 도면이다.

도 3은 파일의 입출력 빈도값을 카운팅하기 위해 이용되는 해시 테이블을 나타내는 도면이다.

도 4는 본 발명의 다른 실시예에 따른 가상 머신에 대한 입출력 처리 방법을 설명하기 위한 도면이다.

도 5는 파일들의 우선 순위에 따라서 입출력 요청이 저장되는 큐를 나타내는 도면이다.

**발명을 실시하기 위한 구체적인 내용**

[0018] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세한 설명에 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.

[0020] 본 발명은 가상 머신에 대한 입력 또는 출력 파일에 대해 우선 순위를 설정하고, 설정된 우선 순위에 따라 입출력 처리를 수행함으로써, 가상 머신에 대한 입출력 처리 성능을 향상시킬 수 있는 방법을 제안한다.

[0021] 이러한 우선 순위는 일실시예로서, 가상 머신에 대한 입출력 파일의 접근 빈도에 따라서 결정될 수 있다. 다시 말해, 본 발명은 가상 머신에서 입출력되는 파일을 모니터링하고, 가상 머신에서 자주 입출력되는 파일에 대한 우선 순위를 상대적으로 그렇지 못한 파일보다 높게 설정한다.

[0022] 본 발명에 따르면, 우선 순위가 높은 파일의 입출력을 우선적으로 처리함으로써, 가상 머신에 대한 입출력 성능을 향상시킬 수 있다.

[0023] 또한, 본 발명에 따르면, 사용자에게 의해 자주 사용되는 파일의 입출력이 상대적으로 우선 순위가 높은 파일의 입출력이 상대적으로 빨리 처리되고 우선 순위가 낮은 파일의 입출력이 상대적으로 늦게 처리됨으로써, 사용자의 체감 속도도 향상될 수 있다.

[0024] 이하에서, 리눅스 기반의 가상화 환경을 중심으로 본 발명이 설명되나, 본 발명은 리눅스 뿐만 아니라 윈도우, 맥 OS 등 다양한 운영체제 기반의 가상화 환경이나 베어메탈 가상화 환경에서도 적용될 수 있다.

[0025] 이하에서, 본 발명에 따른 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다.

[0027] 도 1은 본 발명의 일실시예에 따른 가상화 시스템을 설명하기 위한 도면이다.

[0028] 도 1을 참조하면, 하이퍼바이저(121)는 호스트 머신(111) 상에서 구동되며, 하이퍼바이저(121) 상에 적어도 하나의 가상 머신(131 내지 133)이 구동된다. 호스트 머신(111)에서는 호스트 OS가 구동되며, 각각의 가상 머신에서는 서로 다른 게스트 OS가 구동될 수 있다. 도 1에서는 3개의 가상 머신(131 내지 133)이 구동되는 실시예가 도시된다.

[0029] 가상 머신(131 내지 133)에서 파일의 입출력 요청이 발생하면, 가상 머신 별로 할당된 가상 디스크에 저장된 파

일이 가상 머신으로 리드(read)되거나 또는 가상 디스크에 파일이 라이트(write)될 수 있다. 이러한 가상 디스크는, 호스트 머신(111)의 물리적인 스토리지가 논리적으로 할당된 디스크이다.

- [0030] 본 발명에 따른 하이퍼바이저(121)는 가상 머신(131 내지 133)으로부터, 가상 머신(131 내지 133)에서 입력 입력 또는 출력되는 파일의 메타데이터를 제공받아, 가상 머신(131 내지 133)에서 입력 또는 출력되는 파일의 우선 순위 정보를 생성한다. 메타데이터(metadata)는 게스트 OS(커널)의 파일시스템 계층으로부터 제공될 수 있다.
- [0031] 하이퍼바이저(121)는 우선 순위 정보를 호스트 OS(커널)로 전달하고, 호스트 머신(111) 즉, 호스트 OS는 우선 순위 정보에 따라서, 가상 머신(131 내지 133)에 대한 파일 입출력 요청을 처리한다.
- [0032] 다시 말해, 호스트 머신(111)은 우선 순위가 높은 파일에 대한 입력 또는 출력을 상대적으로 빠르게 처리한다. 예컨대, 제1가상 머신(131)에서 제1파일 및 제2파일의 입력 요청이 발생하였으며, 제1파일의 우선 순위가 제2파일보다 높다면, 제2파일의 입력 요청이 제1파일의 입력 요청보다 먼저 발생하였다더라도, 호스트 머신(111)은 제1 파일에 대한 입력 요청을 제2파일보다 우선적으로 처리한다.
- [0033] 이하, 하이퍼바이저 및 호스트 머신의 입출력 처리 방법을 보다 상세히 설명하기로 한다.
- [0035] 도 2는 본 발명의 일실시예에 따른 가상 머신에 대한 입출력 처리 방법을 설명하기 위한 도면으로서, 하이퍼바이저에서 수행되는 입출력 처리 방법을 설명하기 위한 도면이다. 도 3은 파일의 입출력 빈도값을 카운팅하기 위해 이용되는 해시 테이블을 나타내는 도면이다.
- [0036] 본 발명에 따른 하이퍼바이저는 가상 머신에 대한 입력 또는 출력 파일의 메타데이터로부터, 입력 또는 출력 파일에 대한 식별값을 추출(S310)한다.
- [0037] 여기서, 식별값은 가상 머신에서 입출력되는 모든 파일을 식별해주는 값을 의미하며, 일실시예로서 리눅스 기반의 가상화 시스템에서는 입출력되는 파일의 메타데이터에 포함된 inode 번호나 섹터 번호(sector number) 등이 식별값으로 이용될 수 있다.
- [0038] 본 발명에 따른 하이퍼바이저는 식별값을 이용하여, 입력 또는 출력 파일 각각에 대한 우선 순위 정보를 생성(S320)하며, 가상 머신에 대한 입력 또는 출력이 발생할 때마다, 메타데이터를 제공받아 우선 순위 정보를 갱신할 수 있다.
- [0039] 일실시예로서, 하이퍼바이저는 식별값을 이용하여, 입력 또는 출력 파일 각각에 대한 입출력 빈도값을 카운팅하고, 입출력 빈도값에 따라서, 우선 순위 정보를 생성한다. 입출력되는 빈도가 높을수록, 보다 높은 우선순위가 파일에 할당될 수 있다.
- [0040] 그리고 하이퍼바이저는 파일의 입출력이 발생할 때마다 추출되는 식별값을 카운팅하여 입출력 빈도값을 카운팅하여 우선 순위 정보를 갱신할 수 있다. 다시 말해 실시간으로 입출력 빈도값이 갱신되며, 입출력 빈도값의 갱신에 따라서 실시간으로 우선 순위 정보 역시 갱신된다.
- [0041] 하이퍼바이저는 다양한 방식으로 입출력 빈도값을 카운팅할 수 있으며, 예컨대 미리 설정된 테이블을 이용하여 입출력 빈도값을 카운팅할 수 있다. 하이퍼바이저는 미리 설정된 테이블의 행에, 식별값을 대응시키고, 대응된 행에, 식별값에 대한 노드를 생성할 수 있다. 노드는, 식별값의 개수에 따라서 카운팅되는 입출력 빈도값을 포함하며, 하이퍼바이저는 노드에 포함된 입출력 빈도값을 이용하여 우선 순위 정보를 생성할 수 있다.
- [0042] 일실시예로서, 하이퍼바이저는 도 3과 같은 해시 테이블(hash table)을 이용하여 입출력 빈도값을 카운팅할 수 있다. 도 3에서는 11개의 행을 포함하는 테이블에 노드가 생성되는 실시예가 도시되어 있다.
- [0043] 테이블을 구성하는 행의 개수가 증가할수록 입출력 빈도값에 액세스하는 시간이 증가하므로, 본 발명에 따른 하이퍼바이저는 테이블의 행을 일정 개수로 제한하고 식별값인 inode 번호를 전체 행의 개수로 나누었을 때의 나머지를 이용해 식별값과 테이블의 행을 대응시킨다.
- [0044] 도 3(a)와 같이, inode 번호가 19이고 행의 개수가 11개일 경우, 19를 11로 나누면 나머지가 8이므로, inode 번호 19에 대응되는 행은 8번 행으로 결정된다. 그리고 도 3(b)와 같이 inode 번호가 118일 경우에도 11로 나누면 나머지가 8이므로, inode 번호 118에 대응되는 행은 8번 행으로 결정된다.
- [0045] 도 3(a)은 입출력된 파일이 동일하고 또한 연속으로 이루어진 경우를 나타내며, inode 19번에 대응되는 파일에 대한 입출력 빈도값은 2이므로, 하이퍼바이저는 8번 행에 식별값 19와 입출력 빈도값 2를 포함하는 제1노드

(310)를 생성한다.

- [0046] 이후 도 3(b)와 같이, inode 118번에 대응되는 파일의 입출력이 수행되면, 하이퍼바이저는 제1노드(310) 옆에, 식별값 118과 입출력 빈도값 1을 포함하는 제2노드(320)를 생성한다. 만일 이후에 재차 inode 118번에 대응되는 파일의 입출력이 수행되면, 하이퍼바이저는 제2노드(320)의 입출력 빈도값을 2로 갱신한다.
- [0047] 한편, 서로 다른 가상 머신에서 동일한 식별값을 갖는 파일의 입출력이 발생할 수 있으므로, 하이퍼바이저는 가상 머신에 대한 인덱스값을 더 포함하는 노드를 생성할 수 있다.
- [0048] 다시 도 2로 돌아가, 하이퍼바이저는 단계 S320에서, 우선순위 정보를 생성하기 위해 T-score나 Z-score와 같은 표준 점수(standard score)를 이용할 수 있다.
- [0049] 이를 위해 하이퍼바이저는 입출력 빈도값에 대한 평균 및 표준편차를 이용하여, 식별값에 대한 표준 점수를 생성하고, 생성된 표준 점수에 따라서, 우선 순위 정보를 생성할 수 있다. 하이퍼바이저는 [수학식 1]을 이용하여 T-score(t)를 생성할 수 있다.

**수학식 1**

$$t = \frac{x - \mu}{\sigma} \times 10 + 50$$

- [0050]
- [0051] 여기서, x는 테이블에서의 현재 노드의 입출력 빈도값을 나타내며,  $\mu$ 는 모든 노드에 대한 입출력 빈도값의 평균을 나타낸다. 그리고  $\sigma$ 는 모든 노드에 대한 입출력 빈도값의 표준편차를 나타낸다.
- [0052] 하이퍼바이저는 테이블의 모든 노드에 대해 표준 점수를 생성하며, 식별값은 테이블의 노드 각각에 대응되고, 입출력되는 파일 각각은 식별값에 대응되므로 결국, 하이퍼바이저는 입출력되는 파일 각각에 대해 표준 점수를 생성하게 된다.
- [0053] 하이퍼바이저는 미리 설정된 우선순위와 표준점수 간의 대응 관계에 따라서 최종적으로 입력 또는 출력 파일 각각에 대한 우선 순위 정보를 생성한다.
- [0054] 예컨대, T-score가 45이상인 파일들에 대한 우선순위가 가장 높고, T-score가 35에서 45사이인 파일들에 대한 우선순위가 두번째로 높고, T-score가 35보다 작은 파일들에 대한 우선순위가 가장 낮도록 우선 순위 정보가 생성될 수 있다.
- [0055] 한편, 전술된 실시예에서는 식별값을 통해 입출력 빈도값을 카운팅하여 우선 순위 정보를 생성하는 방법이 설명되었으나, 실시예에 따라서 파일별 우선 순위는 사용자에게 의해 지정될 수도 있을 것이다. 그리고 파일별 우선 순위를 식별하기 위해, 파일에 대한 식별값이 이용될 수 있다.
- [0057] 도 4는 본 발명의 다른 실시예에 따른 가상 머신에 대한 입출력 처리 방법을 설명하기 위한 도면으로서, 호스트 OS에서 수행되는 입출력 처리 방법을 설명하기 위한 도면이다. 그리고 도 5는 파일들의 우선 순위에 따라서 입출력 요청이 저장되는 큐를 나타내는 도면이다.
- [0058] 본 발명에 따른 호스트 OS는 가상 머신에 대한 입력 또는 출력 파일의 우선 순위 정보를 수신 즉, 제공받는다(S410). 호스트 OS는 하이퍼바이저로부터 우선 순위 정보를 제공받을 수 있으며, 우선 순위 정보는 입력 또는 출력 파일에 대한 접근 횟수에 따라서 결정되는 정보이다.
- [0059] 본 발명에 따른 호스트 OS는 가상 머신에 대한 파일 입출력 요청이 발생한 경우, 요청된 파일의 우선 순위 정보에 따라서, 입출력 요청을 우선 순위에 대응되는 서로 다른 큐에 저장(S520)한다.
- [0060] 예컨대, 파일들에 대한 우선 순위가 3가지로 구분되어 있는 경우, 도 5에 도시된 바와 같이 서로 다른 우선 순위에 대응되는 3개의 큐(510, 520, 530)가 이용될 수 있다. 제1큐(510)에는 가장 높은 우선순위를 나타내는 파일들에 대한 입출력 요청이 저장되며, 제2큐(520)에는 두번째로 높은 우선순위를 나타내는 파일들에 대한 입출력 요청이 저장된다. 그리고 제3큐(530)에는 가장 낮은 우선순위를 나타내는 파일들에 대한 입출력 요청이 저장된다.

- [0061] 본 발명에 따른 호스트 OS는 입출력 요청된 파일의 우선 순위에 따라서, 큐에 저장된 입출력 요청을 처리(S530)하며, 우선 순위가 가장 높은 큐에서부터 우선 순위가 가장 낮은 큐의 순서로 입출력 요청을 처리한다.
- [0062] 단계 S530에서 하나의 스레드(thread)가 순환하면서 큐에 할당되고, 이 때, 큐에 할당된 스레드(thread)에 의해 처리되는 입출력 요청의 개수는 우선 순위가 높을수록 많아진다. 예컨대, 도 5에서, 제1큐(510)에 할당된 스레드에 의해 처리되는 입출력 요청의 개수가 제2 및 제3큐(520, 530)에 할당된 스레드에 의해 처리되는 입출력 요청의 개수보다 많고 제2큐(520)에 할당된 스레드에 의해 처리되는 입출력 요청의 개수가 제3큐(530)에 할당된 스레드에 의해 처리되는 입출력 요청의 개수보다 많다. 도 5에서 빗금친 부분은 큐에 할당된 스레드에 의해 처리되는 입출력 요청의 개수를 나타낸다.
- [0063] 따라서, 도 5에서 제1큐(510)에 저장된 입출력 요청이 가장 먼저 처리되고 제3큐(530)에 저장된 입출력 요청이 가장 늦게 처리되며, 제1큐(510)에 저장된 8개의 입출력 요청이 처리된 이후 제2큐(520)에 저장된 입출력 요청이 처리된다. 그리고 제2큐(520)에 저장된 5개의 입출력 요청이 처리된 이후, 제3큐(530)에 저장된 2개의 입출력 요청이 처리된다.
- [0064] 호스트 OS는 가장 낮은 우선 순위에 대응된 입출력 요청이 처리된 이후, 즉, 제3큐(530)에 저장된 입출력 요청이 처리된 이후, 다시 가장 높은 우선 순위에 대응되는 입출력 요청, 즉 제1큐(510)에 저장된 입출력 요청을 처리한다. 다시 말해, 스레드가 제1 내지 제3큐(510 내지 530)에 순환하여 할당되며, 제1 내지 제3큐(510 내지 530)에 저장된 입출력 요청은 순환하며 처리된다.
- [0065] 이러한 본 발명에 따른 방법은 호스터 커널 내에 별도의 모듈에서 수행될 수 있으며, 이러한 모듈을 RTM(Rotated Thread Module) 모듈로 명명한다면, 호스터 커널은 RTM 모듈에 구현된 함수를 호출할 수 있도록 커스터마이징될 수 있다.
- [0066] [표 1]은 동일한 조건에서 400개의 파일을 대상으로 랜덤하게 500회의 쓰기 동작을 수행한 경우에 소요된 시간을 측정한 표이며, [표 2]는 동일한 조건에서 100개의 파일을 대상으로 랜덤하게 500회의 읽기 동작을 수행한 경우에 소요된 시간을 측정한 표이다. “RTM on” 은 본 발명에 따른 측정 결과를 나타내며, “RTM off” 는 기존 시스템에 따른 측정 결과를 나타낸다.

**표 1**

단위(sec)	RTM on		RTM off	
	1MB	4KB	1MB	4KB
전체 시간	4.81 (25%↓)	1.48 (41%↓)	6.4	2.5
평균 시간	0.0096	0.0029	0.0128	0.005

[0067]

**표 2**

단위(sec)	RTM on		RTM off	
	1MB	4KB	1MB	4KB
전체 시간	2.45 (11%↓)	1.12 (20%↓)	2.74	1.4
평균 시간	0.0087	0.0044	0.01	0.0057

[0068]

[0069] [표 1] 및 [표 2]에 나타나 있듯이, 본 발명에 따라서 파일 입력 또는 출력이 처리될 경우, 가상 머신에 대한



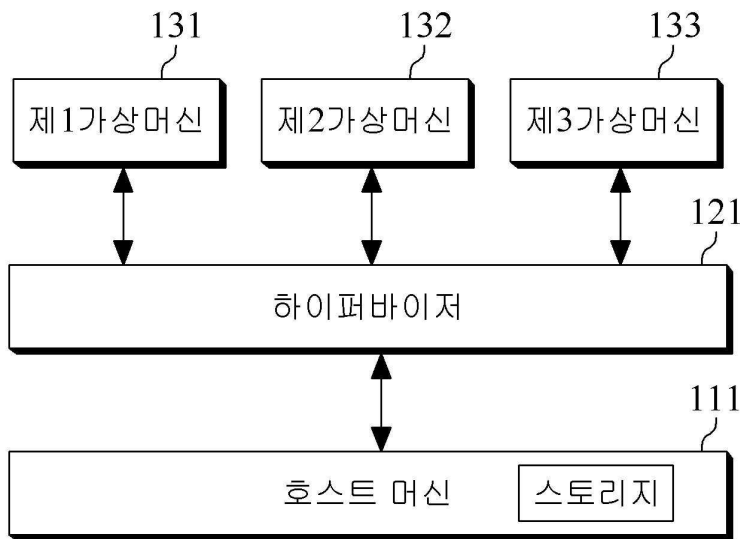
입출력 시간 소요가 줄어드는 것을 확인할 수 있으며, 상대적으로 작은 크기의 파일에 대한 입출력 시간 소요가 많이 줄어드는 것을 확인할 수 있다.

[0071] 앞서 설명한 기술적 내용들은 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 실시예들을 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. 하드웨어 장치는 실시예들의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

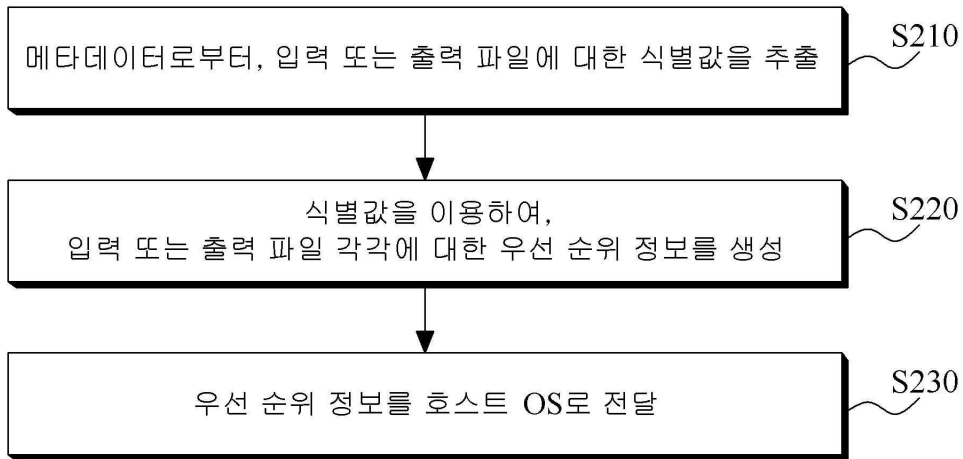
[0073] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다. 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고 할 것이다.

**도면**

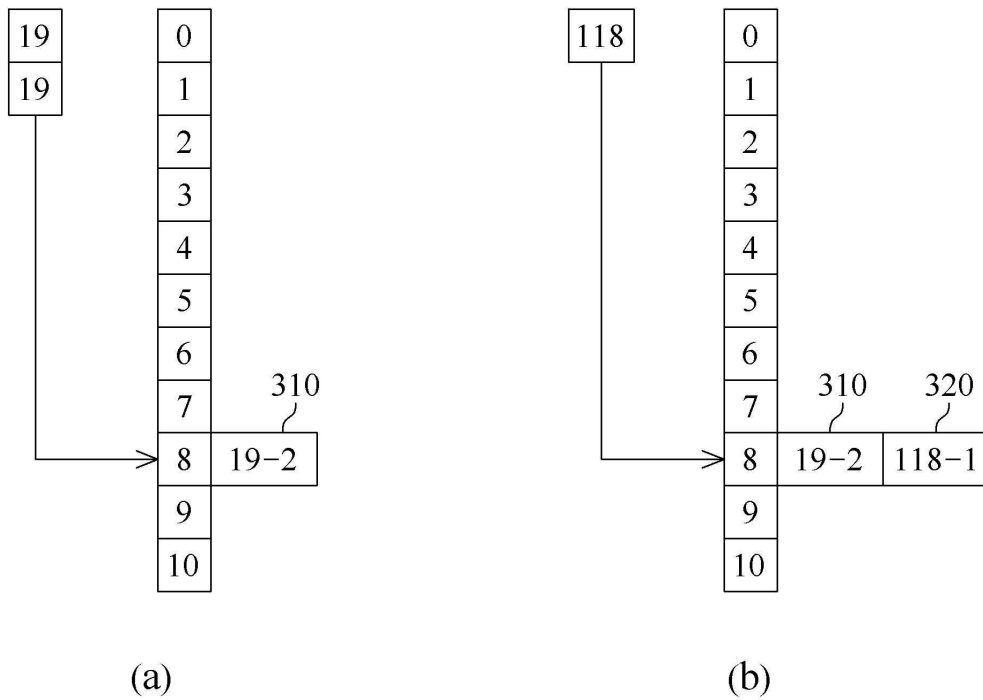
**도면1**



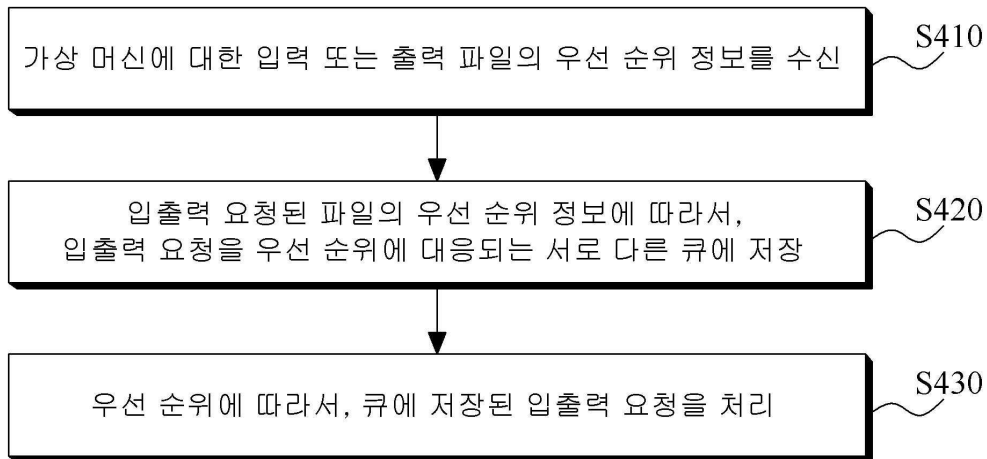
도면2



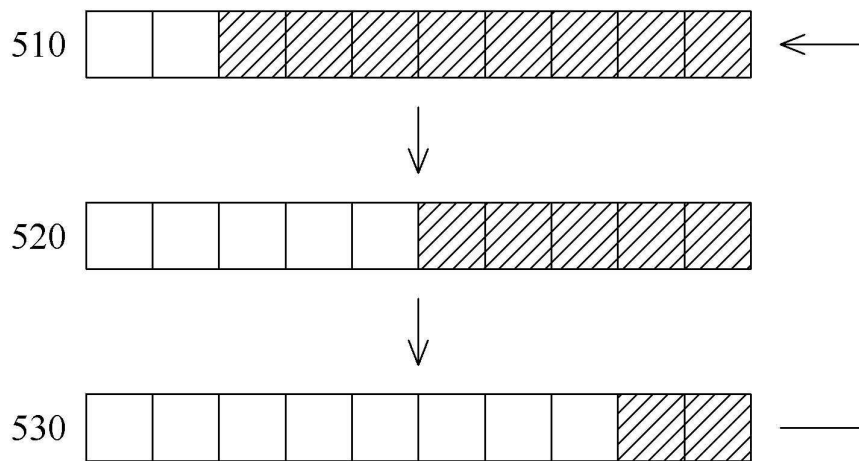
도면3



도면4



도면5



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 1

【변경전】

상기 입출력 빈도값에 따라서, 상기 우선 순위 정보를 생성하는 단계

【변경후】

상기 입출력 빈도값에 따라서, 우선 순위 정보를 생성하는 단계