

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0089858

(43) 공개일자 2025년06월19일

(51) 국제특허분류(Int. Cl.)

G06F 3/06 (2006.01) G06F 12/0888 (2016.01)

(52) CPC특허분류

G06F 3/0659 (2013.01)

G06F 12/0888 (2013.01)

(21) 출원번호 10-2023-0179619

(22) 출원일자 2023년12월12일

심사청구일자 2023년12월12일

(71) 출원인

(주) 글루시스

경기도 안양시 동안구 시민대로327번길 11-31 ,
5층(관양동, 파낙스알앤디센터)

서강대학교산학협력단

서울특별시 마포구 백범로 35 (신수동, 서강대학교)

(72) 발명자

박여현

서울특별시 서초구 매향로 64, 5층(양재동, 엠제이)

박준혁

경기도 안양시 동안구 동편로 135, 412동 804호
(관양동, 동편마을)

(뒷면에 계속)

(74) 대리인

수안특허법인, 민영준

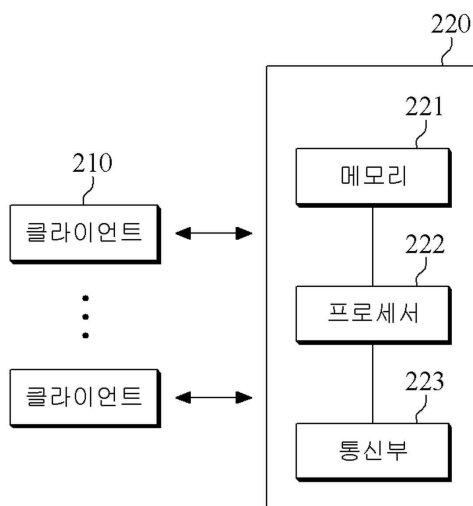
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 스토리지 서버 및 스토리지 서버의 입출력 처리 방법

(57) 요약

부하 불균형 문제와 입출력 요청 서비스에 대한 공정성 문제를 해결할 수 있는 스토리지 서버 및 스토리지 서버의 입출력 처리 방법이 개시된다. 개시된 스토리지 서버의 입출력 처리 방법은 복수의 클라이언트로부터 입출력 요청을 수신하는 단계; 상기 스토리지 서버의 프로세서 코어별 사용량을 이용하여, 상기 입출력 요청의 처리를 위한 프로세서 코어를 결정하는 스케줄링을 수행하는 단계; 및 상기 입출력 요청에 대한 서비스를, 상기 클라이언트로 제공하는 단계를 포함한다.

대표도 - 도2



(52) CPC특허분류

G06F 3/0604 (2013.01)

G06F 3/061 (2013.01)

G06F 2212/1024 (2013.01)

(72) 발명자

박정환

서울특별시 강동구 양재대로 1716, 1동 406호 (고덕동, 대우아파트)

김경표

광주광역시 동구 용산3길 17(용산동, 모아엘가 에듀파크)

박성순

경기도 군포시 수리산로 244, 995동 901호(산본동, 한양백두아파트)

김영재

서울특별시 서초구 잠원로12길 4, 101동 1808호(잠원동, 잠원현대아파트)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711193184
과제번호	2020-0-00104-004
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	차세대엡지컴퓨팅시스템기술개발(R&D)
연구과제명	저지연 입출력 집약적 엡지 데이터 처리를 위한 스토리지 모듈 기술 개발
기 여 율	1/1
과제수행기관명	(주)글루시스
연구기간	2023.01.01 ~ 2023.12.31

명세서

청구범위

청구항 1

스토리지 서버의 입출력 처리 방법에 있어서,

복수의 클라이언트로부터 입출력 요청을 수신하는 단계;

상기 스토리지 서버의 프로세서 코어별 사용량을 이용하여, 상기 입출력 요청의 처리를 위한 프로세서 코어를 결정하는 스케줄링을 수행하는 단계; 및

상기 입출력 요청에 대한 서비스를, 상기 클라이언트로 제공하는 단계를 포함하는 입출력 처리 방법.

청구항 2

제 1항에 있어서,

상기 스케줄링을 수행하는 단계는

상기 프로세서 코어 각각의 사용량을 모니터링하는 단계;

상기 모니터링된 사용량 중 최대 사용량과, 최소 사용량의 비율값을 계산하는 단계; 및

상기 비율값이 제1임계값 이하인 경우, 상기 스케줄링을 수행하는 단계를 포함하는 입출력 처리 방법.

청구항 3

제 2항에 있어서,

상기 스케줄링을 수행하는 단계는

상기 프로세서 코어 중, 상기 사용량이 제2임계값 이상인 과부하 프로세서 코어에 대해, 상기 스케줄링을 수행하는

입출력 처리 방법.

청구항 4

제 3항에 있어서,

상기 스케줄링을 수행하는 단계는

상기 과부하 프로세서 코어에 할당된 입출력 요청 중, 입출력 요청 빈도가 제3임계값 이상인 클라이언트의 입출력 요청을, 상기 사용량이 상기 제2임계값보다 작은 프로세서 코어에 할당하는

입출력 처리 방법.

청구항 5

스토리지 서버의 입출력 처리 방법에 있어서,

복수의 클라이언트로부터 입출력 요청을 수신하는 단계;

상기 입출력 요청에 대한 서비스의 공정성에 따라서, 상기 입출력 요청을 처리하는 프로세서 코어에서의, 상기 입출력 요청을 처리하는 순서에 대한 스케줄링을 수행하는 단계; 및

상기 입출력 요청에 대한 서비스를, 상기 클라이언트로 제공하는 단계

를 포함하는 입출력 처리 방법.

청구항 6

제 5항에 있어서,

상기 제1 및 제2스케줄링 중 적어도 하나를 수행하는 단계는

상기 클라이언트 중 어느 하나의 클라이언트의 입출력 요청에 대한 평균 서비스 응답 시간과, 상기 클라이언트 전부의 입출력 요청에 대한 평균 서비스 응답 시간의 비율값을 계산하는 단계; 및

상기 비율값이 제1임계값 이하인 경우, 상기 스케줄링을 수행하는 단계

를 포함하는 입출력 처리 방법.

청구항 7

제 6항에 있어서,

상기 스케줄링을 수행하는 단계는

상기 클라이언트 중, 입출력 요청 빈도가 제2임계값 이상인 제1클라이언트의 입출력 요청의 처리 순서를, 후순위로 조절하는

입출력 처리 방법.

청구항 8

제 7항에 있어서,

상기 제1클라이언트의 입출력 요청의 처리 순서를 후순위로 조절하는 단계는

상기 제1클라이언트의 입출력 요청에 대한 처리 시점에서, 상기 제1클라이언트의 입출력 요청의 순서를, 상기 입출력 요청이 저장된 이벤트 큐에서의 마지막 순서로 조절하는

입출력 처리 방법.

청구항 9

제 8항에 있어서,

상기 제1클라이언트의 입출력 요청의 처리 순서를 후순위로 조절하는 단계는

상기 클라이언트 중, 상기 입출력 요청 빈도가 상기 제2임계값보다 작은 제2클라이언트의 입출력 요청 빈도의 평균값과, 상기 제1클라이언트의 입출력 요청 빈도를 이용하여, 순위 조절 횟수를 계산하는 단계; 및

상기 순위 조절 횟수만큼, 상기 제1클라이언트의 입출력 요청의 처리 순서를 상기 후순위로 조절하는 단계

를 포함하는 입출력 처리 방법.

청구항 10

클라이언트의 입출력 요청을 처리하는 스토리지 서버에 있어서,

메모리; 및

상기 메모리와 전기적으로 연결된 적어도 하나의 프로세서; 및

복수의 클라이언트로부터의 입출력 요청을 수신하는 통신부를 포함하며,

상기 프로세서는

상기 스토리지 서버의 프로세서 코어별 사용량 및 상기 입출력 요청에 대한 서비스의 공정성에 따라서, 제1 및 제2스케줄링 중 적어도 하나를 수행하며,

상기 제1스케줄링은, 상기 입출력 요청의 처리를 위한 프로세서 코어를 결정하는 스케줄링이며,

상기 제2스케줄링은, 상기 입출력 요청을 처리하는 프로세서 코어에서의, 상기 입출력 요청을 처리하는 순서에 대한 스케줄링인

스토리지 서버.

발명의 설명**기술 분야**

[0001] 본 발명은 스토리지 서버 및 스토리지 서버의 입출력 처리 방법에 관한 것으로서, 보다 상세하게는 부하 불균형 문제와 입출력 요청 서비스에 대한 공정성 문제를 해결할 수 있는 스토리지 서버 및 스토리지 서버의 입출력 처리 방법에 관한 것이다.

배경 기술

[0003] LevelDB 및 RocksDB와 같은 내장형 키-값 저장소(KVS)는 데이터 집약적 애플리케이션의 저장 솔루션으로 널리 사용되고 있다. 기존에는 이러한 저장소가 파일 시스템 위에서 실행되어 키-값 쌍을 파일로 변환한 다음 블록 주소로 변환하는 과정에서 I/O 작업에 상당한 성능 오버헤드가 발생했다. 또한 파일시스템 위에서 키-값 저장소를 실행하는 것은 유저-커널 모드 전환, 인터럽트 처리 및 문맥 전환(context switch)과 같은 운영체제 작업을 추가 오버헤드로 가져왔다. NVMe SSD와 같은 초저지연 저장장치의 등장으로 이러한 오버헤드는 과거보다 더 심해졌다.

[0004] 이 문제를 해결하기 위해 KVS의 I/O 경로에서 파일시스템 및 운영체제 구성 요소를 제거하는 다양한 방법이 도입되었다. 그 중 하나가 Intel Storage Performance Development Kit(SPDK)이다. SPDK는 NVMe SSD의 성능을 완전히 극대화하기 위한 소프트웨어 프레임워크를 제공한다. SPDK는 폴링 모드를 사용하는 사용자 레벨의 NVMe 드라이버를 필두로 운영체제와 관련된 전통적인 성능 오버헤드를 줄인다. SPDK는 유저 모드와 커널 모드 간의 문맥 전환을 최소화하고 비용이 많이 드는 인터럽트를 제거했다. 현재, 이러한 SPDK의 장점을 이용해 단일 노드 시스템에서 고성능 KVS 솔루션을 개발하기 위한 노력이 활발히 진행 중이다.

[0005] 한편, 현대 컴퓨팅 환경에서는 분리화 스토리지 저장소(disaggregated storage) 개념이 중심이 되고 있다. 스토리지 분리화 기술은 스토리지 가상화를 활용하여 원격의 디스크나 원격 파일시스템을 클라이언트에게 로컬 디스크 볼륨이나 파일시스템처럼 제공한다. Amazon 클라우드의 EBS(볼륨) 또는 EFS(파일 시스템)가 대표적이다. 이 때, 이러한 분리된 저장소 환경에서 내장형 키-값 저장소를 구축하는 전통적인 방법은 앞서 말한 기존의 방식처럼, 파일 및 블록 레이어 위에서 KVS가 구동되면서 주소 변환 및 커널문맥 전환 오버헤드가 수반되는 방식이다.

[0006] 이에, 최근에는 스토리지 서버의 사용자 공간에서 구동되는 네트워크 기반 키-값 저장소가 개발되고 있다. 네트워크 기반 키-값 저장소는, 스토리지 서버에서 SPDK를 설치하고 그 내부에 직접 키-값 저장소 엔진을 구현한 구조를 택한다. 따라서 클라이언트의 파일 시스템 및 키-값 저장소 스토리지 엔진이 필요하지 않다. 이러한 접근

방식은 앞서 언급한 오버헤드를 초래하는 파일 및 블록 레이어와 같은 운영체제 요소를 우회(bypass)하기 때문에 현저한 성능 향상을 가져온다.

[0007] 네트워크 기반 키-값 저장소는 또한 이러한 SPDK 기반 KVS 솔루션에서 발생하는 부하 불균형 문제를 식별하고 해결하며, 이를 완화하기 위한 로드 밸런싱 알고리즘도 제안해 전체 시스템 성능을 최적화하였다. 네트워크 기반 키-값 저장소는 SPDK의 NVMe over Fabrics (NVMe-oF) 지원도 활용하여 성능과 확장성(scalability)이 향상된 분리화 저장소 아키텍처를 개발할 수 있게 하였다.

[0008] 관련 선행문헌으로 대한민국 공개특허 제2021-0075845호, 제2021-0123236호, 대한민국 등록특허 제10-2471966호가 있다.

발명의 내용

해결하려는 과제

[0010] 본 발명은 부하 불균형 문제와 입출력 요청 서비스에 대한 공정성 문제를 해결할 수 있는 스토리지 서버 및 스토리지 서버의 입출력 처리 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0012] 상기한 목적을 달성하기 위한 본 발명의 일 실시예에 따르면, 복수의 클라이언트로부터 입출력 요청을 수신하는 단계; 스토리지 서버의 프로세서 코어별 사용량을 이용하여, 상기 입출력 요청의 처리를 위한 프로세서 코어를 결정하는 스케줄링을 수행하는 단계; 및 상기 입출력 요청에 대한 서비스를, 상기 클라이언트로 제공하는 단계를 포함하는 스토리지 서버의 입출력 처리 방법이 제공된다.

[0013] 또한 상기한 목적을 달성하기 위한 본 발명의 다른 실시예에 따르면, 복수의 클라이언트로부터 입출력 요청을 수신하는 단계; 상기 입출력 요청에 대한 서비스의 공정성에 따라서, 상기 입출력 요청을 처리하는 프로세서 코어에서의, 상기 입출력 요청을 처리하는 순서에 대한 스케줄링을 수행하는 단계; 및 상기 입출력 요청에 대한 서비스를, 상기 클라이언트로 제공하는 단계를 포함하는 입출력 처리 방법이 제공된다.

[0014] 또한 상기한 목적을 달성하기 위한 본 발명의 또 다른 실시예에 따르면, 메모리; 및 상기 메모리와 전기적으로 연결된 적어도 하나의 프로세서; 및 복수의 클라이언트로부터의 입출력 요청을 수신하는 통신부를 포함하며, 상기 프로세서는 상기 스토리지 서버의 프로세서 코어별 사용량 및 상기 입출력 요청에 대한 서비스의 공정성에 따라서, 제1 및 제2스케줄링 중 적어도 하나를 수행하며, 상기 제1스케줄링은, 상기 입출력 요청의 처리를 위한 프로세서 코어를 결정하는 스케줄링이며, 상기 제2스케줄링은, 상기 입출력 요청을 처리하는 프로세서 코어에서의, 상기 입출력 요청을 처리하는 순서에 대한 스케줄링인, 클라이언트의 입출력 요청을 처리하는 스토리지 서버가 제공된다.

발명의 효과

[0016] 본 발명의 일 실시예에 따르면, 과부하 프로세서 코어에 할당된 입출력 요청 일부가, 부하가 적은 프로세서 코어에 할당됨으로써, 프로세서 코어의 사용률이 균등해질 수 있을뿐만 아니라, 과부하 프로세서 코어에 할당된 입출력 요청의 처리가 지연됨으로써 발생하는 입출력 요청 서비스에 대한 공정성 문제도 해결될 수 있다.

[0017] 또한 본 발명의 일 실시예에 따르면, 입출력 요청 빈도가 낮은 클라이언트의 입출력 요청이 우선적으로 처리됨으로써, 입출력 요청 빈도가 높은 클라이언트의 입출력 요청에 의해, 입출력 요청 빈도가 낮은 클라이언트의 입출력 요청 서비스가 지연되는 현상이 방지될 수 있으므로, 입출력 요청에 따른 서비스 제공의 공정성이 높아질 수 있다.

도면의 간단한 설명

- [0019] 도 1은 스토리지 분리화 환경에서 이용되는 네트워크 기반 키-값 저장소를 설명하기 위한 도면이다.
- 도 2는 본 발명의 일실시예에 따른 스토리지 시스템을 설명하기 위한 도면이다.
- 도 3은 본 발명의 일실시예에 따른 스토리지 서버의 입출력 처리 방법을 설명하기 위한 도면이다.
- 도 4는 본 발명의 다른 실시예에 따른 스토리지 서버의 입출력 처리 방법을 설명하기 위한 도면이다.
- 도 5는 본 발명의 일실시예에 따른 스케줄링 방법을 설명하기 위한 도면이다.
- 도 6은 본 발명의 일실시예에 따른 입출력 요청 서비스의 공정성 개선 효과를 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0020] 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.
- [0021] 이하에서, 본 발명에 따른 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다.
- [0023] 도 1은 스토리지 분리화 환경에서 이용되는 네트워크 기반 키-값 저장소를 설명하기 위한 도면이다.
- [0024] 도 1에 도시된 바와 같이, 스토리지 분리화 환경에서 키-값 저장소(122)는 클라이언트(110)가 아닌 스토리지 서버(120)에 위치한다. 복수의 클라이언트(110)는 응용(111, application) 및 NVMe-oF 드라이버(112)를 포함하며, 스토리지 서버(120)는 NVMe-oF 타겟(121), 키-값 저장소(122), NVMe 드라이버(123) 및 NVMe SSD(124)를 포함한다. NVMe-oF 타겟(121), 키-값 저장소(122), NVMe 드라이버(123)는 SPDK를 이용하여 구현될 수 있다.
- [0025] 클라이언트의 응용(111)은 키-값에 대한 읽기 또는 쓰기 요청과 같은 입출력(I/O) 요청을 NVMe-oF 드라이버(112)로 전달하고, NVMe-oF 드라이버(112)는 입출력 요청을 NVMe 명령어로 변환한다. 즉, 입출력 요청은 클라이언트의 키-값 저장소와 커널 영역의 파일 시스템을 거치지 않고, 직접 NVMe-oF 드라이버(112)로 전달된다. 입출력 요청이 클라이언트의 파일 시스템을 통과하여, NVMe-oF 드라이버(112)에 의해 NVMe 명령어로 변환되며, 키-값 저장소(122)는 SPDK에 기반하여 스토리지 서버(120)의 사용자 공간에 구현되기 때문에, 커널 I/O 스택의 오버헤드가 감소할 수 있다.
- [0026] NVMe-oF 드라이버(112)는 입출력 요청을, 입출력 요청에 대응되는 NVMe 명령어로 변환한 후 스토리지 서버(120)로 전송한다.
- [0027] 스토리지 서버(120)로 전송된 NVMe 명령어는 NVMe-oF 타겟(121)에 의해 키-값 저장소(122)로 전달되며, 키-값 저장소(122)는 NVMe 명령어를 블록 I/O 요청으로 변환하여 NVMe 드라이버(123)으로 전달한다. 그리고 NVMe SSD(124)가 블록 I/O 요청에 따라 데이터를 저장하거나 읽어옴으로써, 입출력 요청에 대한 서비스가 클라이언트(110)로 제공될 수 있다.
- [0028] 이 때, 스토리지 서버는 클라이언트의 입출력 요청을, 요청 순서에 따라 순차적으로 처리하며, 따라서 복수의 클라이언트로부터 입출력 요청이 생성될 경우, 입출력 요청 서비스에 대한 불공정함이 발생할 수 있다. 즉, 특정 클라이언트로부터 많은 입출력 요청이 생성될 경우, 다른 클라이언트의 입출력 요청에 대한 처리가 지연되어, 입출력 요청에 따른 서비스 제공에 차별이 발생할 수 있다.
- [0029] 또한 스토리지 서버의 특정 프로세서 코어에서의 입출력 요청 처리 부하가 증가되어, 프로세서 코어 사이의 부하 불균형 문제가 발생할 수 있다.
- [0030] 이에 본 발명은 부하 불균형 문제와 입출력 요청 서비스에 대한 공정성 문제를 해결할 수 있는 스토리지 서버 및 스토리지 서버의 입출력 처리 방법을 제안한다. 본 발명의 일실시예는 스케줄링을 통해 부하 불균형 문제와 입출력 요청 서비스에 대한 공정성 문제를 해결한다.
- [0032] 도 2는 본 발명의 일실시예에 따른 스토리지 시스템을 설명하기 위한 도면이다.

- [0033] 도 2를 참조하면 본 발명의 일실시예에 따른 스토리지 시스템은 복수의 클라이언트(210)와 스토리지 서버(220)를 포함한다.
- [0034] 복수의 클라이언트(210)는 각각 스토리지 서버(220)로 데이터의 입력 또는 출력을 요청하며, 스토리지 서버(220)는 클라이언트(210)의 입출력 요청에 대한 서비스를 클라이언트(210)로 제공한다. 즉, 스토리지 서버(220)는 클라이언트(210)의 입력 요청된 데이터를 저장하고, 출력 요청된 데이터를 클라이언트(210)로 전송한다. 스토리지 서버(220)는 메모리(221), 적어도 하나의 프로세서(222) 및 통신부(223)를 포함한다.
- [0035] 통신부(223)는 복수의 클라이언트(210)로부터 입출력 요청을 수신한다. 통신부(223)는 입력 요청된 데이터를 클라이언트(210)로부터 수신하고, 출력 요청된 데이터를 클라이언트(210)로 전송한다.
- [0036] 프로세서(222)는 메모리(221)와 전기적으로 연결되며, 클라이언트(210)의 입출력 요청에 응답하여, 입출력 서비스를 제공하기 위한 일련의 과정을 수행한다. 프로세서(222)는 일실시예로서, 멀티코어 프로세서 또는 매니코어 프로세서일 수 있다.
- [0037] 프로세서(222)는 스토리지 서버(220)의 프로세서 코어별 사용량 및 클라이언트(210)의 입출력 요청에 대한 서비스의 공정성에 따라서, 제1 및 제2스케줄링 중 적어도 하나를 수행한다. 여기서, 제1스케줄링은, 클라이언트(210)의 입출력 요청의 처리를 위한 프로세서 코어를 결정하는 스케줄링이며, 제2스케줄링은, 입출력 요청을 처리하는 프로세서 코어에서의, 입출력 요청을 처리하는 순서에 대한 스케줄링이다.
- [0038] 프로세서(222)는 제1 및 제2스케줄링을 모두 수행하거나, 제1 및 제2스케줄링 중 하나를 선택적으로 수행할 수 있다. 프로세서(222)는 프로세서 코어별 사용량을 이용해 미리 설정된 제1조건이 만족되는지 여부를 판단하고, 제1조건이 만족되는 경우, 제1스케줄링을 수행할 수 있다. 또한 입출력 요청에 대한 서비스의 공정성이 제2조건을 만족하는 경우, 제2스케줄링을 수행할 수 있다. 그리고 전술된 제1 및 제2조건이 모두 만족되는 경우, 프로세서(222)는 제1 및 제2스케줄링을 모두 수행할 수 있다. 여기서, 제1조건은 프로세서 코어 사이에 부하 불균형이 발생한 조건이며, 제2조건은 입출력 요청에 대한 서비스 제공에 차별이 발생하여, 입출력 요청에 대한 서비스 제공이 공정하지 못한 조건이다.
- [0040] 도 3은 본 발명의 일실시예에 따른 스토리지 서버의 입출력 처리 방법을 설명하기 위한 도면이다.
- [0041] 도 3을 참조하면, 본 발명의 일실시예에 따른 스토리지 서버는 복수의 클라이언트로부터 입출력 요청을 수신(S310)하고, 스토리지 서버의 프로세서 코어별 사용량을 이용하여, 입출력 요청의 처리를 위한 프로세서 코어를 결정하는 스케줄링을 수행(S320)한다.
- [0042] 단계 S320에서 스토리지 서버는 전술된 제1조건이 만족되는지 여부를 판단하며, 제1조건이 만족되는지 여부를 판단하기 위해, 프로세서 코어 각각의 사용량을 모니터링한다. 모니터링은 미리 설정된 모니터링 구간 단위로 수행될 수 있으며, 스토리지 서버는 모니터링된 사용량 중 최대 사용량과, 최소 사용량의 비율값을 계산한다. 즉, 프로세서 코어별로 최대 사용량과, 최소 사용량의 비율값을 계산하는 것이 아니라, 전체 프로세서 코어의 사용량 중 최대 사용량과, 최소 사용량의 비율값을 계산한다. 그리고 스토리지 서버는 비율값이 제1임계값 이하인 경우, 전술된 제1조건이 만족된 것으로 판단하여, 프로세서 코어를 결정하는 스케줄링을 수행한다.
- [0043] 일실시예로서, 스토리지 서버는 프로세서 코어 중, 사용량이 제2임계값 이상인 과부하 프로세서 코어에 대해, 스케줄링을 수행할 수 있다. 그리고 과부하 프로세서 코어에 할당된 입출력 요청 중, 입출력 요청 빈도가 제3임계값 이상인 클라이언트의 입출력 요청을, 사용량이 제2임계값보다 작은 프로세서 코어에 할당하는 스케줄링을 수행할 수 있다. 즉, 스토리지 서버는 입출력 요청 빈도 높은 클라이언트의 입출력 요청이 과부하 프로세서 코어에 할당된 경우, 입출력 요청 빈도 높은 클라이언트의 입출력 요청이 부하가 적은 프로세서 코어에서 처리될 수 있도록 스케줄링을 수행한다.
- [0044] 그리고 스토리지 서버는 입출력 요청에 대한 서비스를, 클라이언트로 제공(S330)한다. 스토리지는 스케줄링을 수행하여, 클라이언트에 대한 입출력 요청을 처리하며, 입력 요청된 데이터를 저장하고, 출력 요청된 데이터를 클라이언트로 전송하는 서비스를 제공한다.
- [0045] 본 발명의 일실시예에 따르면, 과부하 프로세서 코어에 할당된 입출력 요청 일부가, 부하가 적은 프로세서 코어에 할당됨으로써, 프로세서 코어의 사용률이 균등해질 수 있을뿐만 아니라, 과부하 프로세서 코어에 할당된 입출력 요청의 처리가 지연됨으로써 발생하는 입출력 요청 서비스에 대한 공정성 문제도 해결될 수 있다.

- [0047] 도 4는 본 발명의 다른 실시예에 따른 스토리지 서버의 입출력 처리 방법을 설명하기 위한 도면이며, 도 5는 본 발명의 일실시예에 따른 스케줄링 방법을 설명하기 위한 도면이다.
- [0048] 도 4를 참조하면 본 발명의 일실시예에 따른 스토리지 서버는, 복수의 클라이언트로부터 입출력 요청을 수신(S410)하고, 클라이언트의 입출력 요청에 대한 서비스의 공정성에 따라서, 입출력 요청을 처리하는 스토리지 서버의 프로세서 코어에서의, 입출력 요청을 처리하는 순서에 대한 스케줄링을 수행(S420)한다.
- [0049] 단계 S420에서 스토리지 서버는 전송된 제2조건이 만족되는지 여부를 판단하며, 제2조건이 만족되는지 여부를 판단하기 위해, 클라이언트의 입출력 요청에 대한 서비스 응답 시간을 모니터링한다. 여기서, 서비스 응답 시간은 클라이언트의 입출력 요청 시점으로부터, 입출력 요청된 서비스가 제공되는 시점까지의 시간을 의미한다.
- [0050] 그리고 스토리지 서버는 클라이언트 중 어느 하나의 클라이언트의 입출력 요청에 대한 평균 서비스 응답 시간과, 클라이언트 전부의 입출력 요청에 대한 평균 서비스 응답 시간의 비율값을 계산한다. 예컨대, 스토리지 서버가 4개의 클라이언트로 서비스를 제공하는 환경인 경우, 스토리지 서버는 4개 중 1개의 클라이언트만이 스토리지 서버로 입출력을 요청한 경우의 평균 서비스 응답 시간과, 4개 모두의 클라이언트가 함께 스토리지 서버로 입출력을 요청한 경우의 평균 서비스 응답 시간의 비율값을 계산할 수 있다. 스토리지 서버는 미리 설정된 모니터링 구간에서 측정된 서비스 응답 시간의 개수를 이용해, 평균 서비스 응답 시간을 계산할 수 있다.
- [0051] 그리고 스토리지 서버는 비율값이 제4임계값 이하인 경우, 전송된 제2조건이 만족된 것으로 판단하여, 입출력 요청을 처리하는 순서에 대한 스케줄링을 수행할 수 있다.
- [0052] 일실시예로서, 스토리지 서버는 클라이언트 중, 입출력 요청 빈도가 제5임계값 이상인 제1클라이언트의 입출력 요청의 처리 순서를, 후순위로 조절하는 스케줄링을 수행할 수 있다. 즉 스토리지 서버는 입출력 요청 빈도가 높은 제1클라이언트의 입출력 요청보다, 입출력 요청 빈도가 제5임계값보다 낮은 제2클라이언트의 입출력 요청이 우선적으로 처리되도록 스케줄링을 수행할 수 있다.
- [0053] 일실시예로서, 클라이언트의 입출력 요청은 이벤트로 변환되어 이벤트 큐에 삽입되고, 스토리지 서버는 이벤트가 이벤트 큐에 삽입된 순서대로, 입출력 요청을 처리할 수 있는데, 이 때 스토리지 서버는 제1클라이언트의 입출력 요청에 대한 처리 시점에서, 제1클라이언트의 입출력 요청의 순서를, 입출력 요청이 저장된 이벤트 큐에서의 마지막 순서로 조절함으로써, 제1클라이언트의 입출력 요청의 처리 순서를, 후순위로 조절할 수 있다.
- [0054] 그리고 스토리지 서버는 클라이언트의 입출력 요청에 대한 서비스를, 클라이언트로 제공(S430)한다.
- [0055] 따라서, 본 발명의 일실시예에 따르면, 입출력 요청 빈도가 낮은 클라이언트의 입출력 요청이 우선적으로 처리됨으로써, 입출력 요청 빈도가 높은 클라이언트의 입출력 요청에 의해, 입출력 요청 빈도가 낮은 클라이언트의 입출력 요청 서비스가 지연되는 현상이 방지될 수 있으므로, 입출력 요청에 따른 서비스 제공의 공정성이 높아질 수 있다.
- [0056] 한편, 제1클라이언트의 입출력 요청을 개수 제한없이 후순위로 처리하는 것은, 제1클라이언트에 대한 역차별이 될 수 있으므로, 스토리지 서버는 단계 S420에서, 미리 설정된 순위 조절 횟수만큼, 제1클라이언트의 입출력 요청의 처리 순서를 후순위로 조절할 수 있다.
- [0057] 예컨대, 도 5(a)와 같이, 제1클라이언트의 4개 이벤트(511 내지 514)가, 제2클라이언트들의 6개 이벤트(520) 보다 먼저 이벤트 큐에 삽입된 상태이며, 순위 조절 횟수가 3인 경우, 스토리지 서버는 도 5(b)와 같이, 제1클라이언트의 4개 이벤트(511 내지 514) 중, 3개 이벤트(511 내지 513)의 처리 순서를 후순위로 조절할 수 있다.
- [0058] 스토리지 서버는 클라이언트 중, 입출력 요청 빈도가 제2임계값보다 작은 제2클라이언트의 입출력 요청 빈도의 평균값과, 제1클라이언트의 입출력 요청 빈도를 이용하여, 순위 조절 횟수를 계산할 수 있다. 스토리지 서버는 일실시예로서, [수학식 1]을 이용하여, 순위 조절 횟수를 계산할 수 있으며, 순위 조절 횟수는 제1클라이언트의 입출력 요청 빈도에 비례하도록 결정될 수 있다.

수학식 1

$$Backwardings(B) = \frac{f_{intensity}(flow_{high}^i)}{I_{sparse_avg}} \alpha$$

여기서, $Backwardings(B)$ 는 순위 조절 횟수, I_{sparse_avg} 는 제2클라이언트의 입출력 요청 빈도의 평균값, $f_{intensity}(flow_{high}^i)$ 는 제1클라이언트의 입출력 요청 빈도, α 는 가중치를 나타낸다.

스토리지 서버는 미리 설정된 모니터링 구간에서 클라이언트들의 입출력 요청 빈도를 카운팅하여, 제1 및 제2클라이언트를 결정할 수 있으며, 제2클라이언트의 입출력 요청 빈도의 평균값 역시 계산할 수 있다.

도 6은 본 발명의 일실시예에 따른 입출력 요청 서비스의 공정성 개선 효과를 설명하기 위한 도면이다.

도 6에서 inter-core는 전술된 제1스케줄링이 수행되는 경우, intra-core는 전술된 제2스케줄링이 수행되는 경우, hybrid는 제1 및 제2스케줄링이 모두 수행되는 경우, baseline은 제1 및 제2스케줄링이 모두 수행되지 않는 경우를 나타내며, 도 6은 전술된 4가지 경우에서의 송수신되는 데이터의 스루풋과 서비스 공정성을 측정한 도면이다.

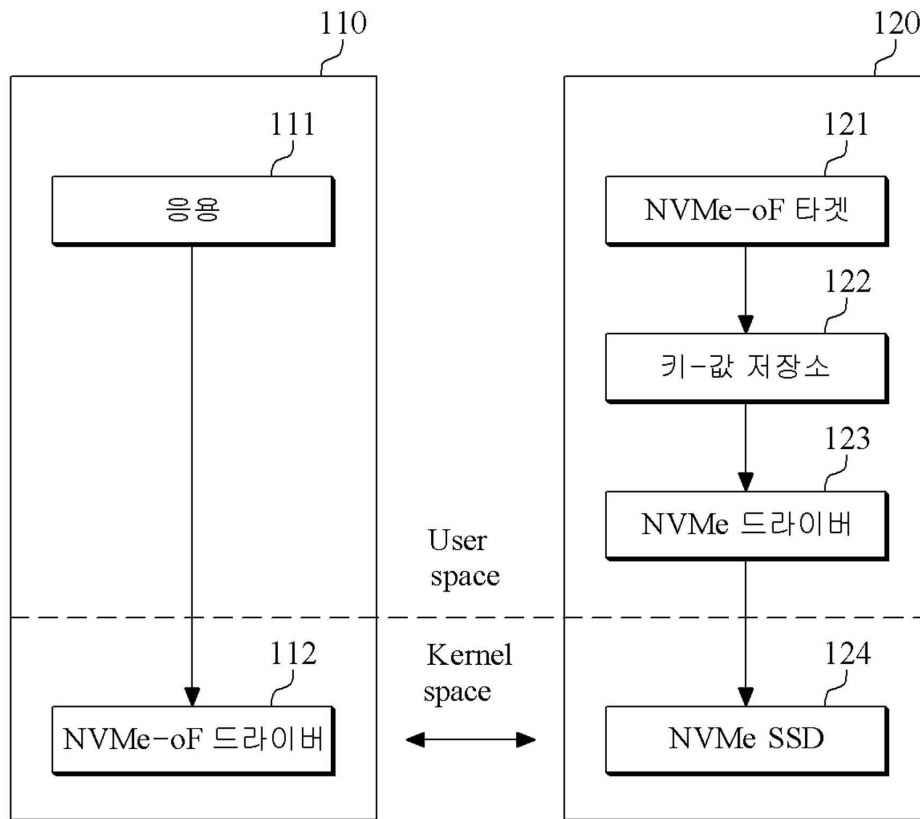
도 6에서 공정성을 나타내는 값이 높을수록 서비스 공정성이 높음을 의미하며, 도 6에 도시된 바와 같이, 본 발명의 일실시예에 따른 스케줄링이 수행되는 경우에 입출력 요청 서비스에 대한 공정성이 개선됨을 알 수 있다. 여기서 공정성을 나타내는 값은, 전술된 평균 서비스 응답 시간의 비율값에 대응된다.

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.

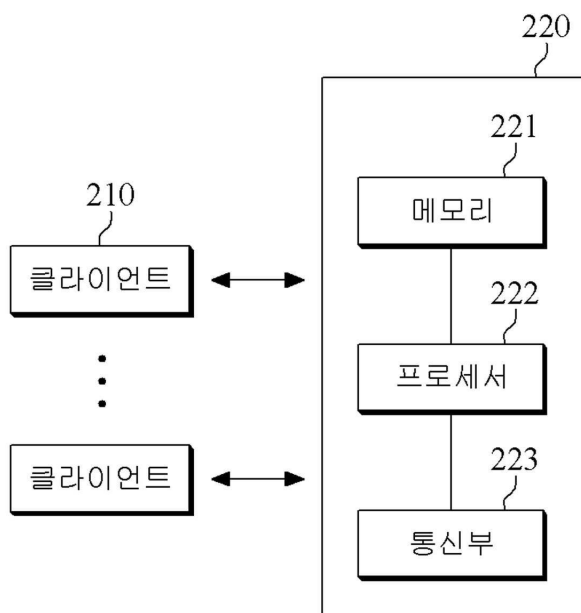
이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다. 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고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

도면

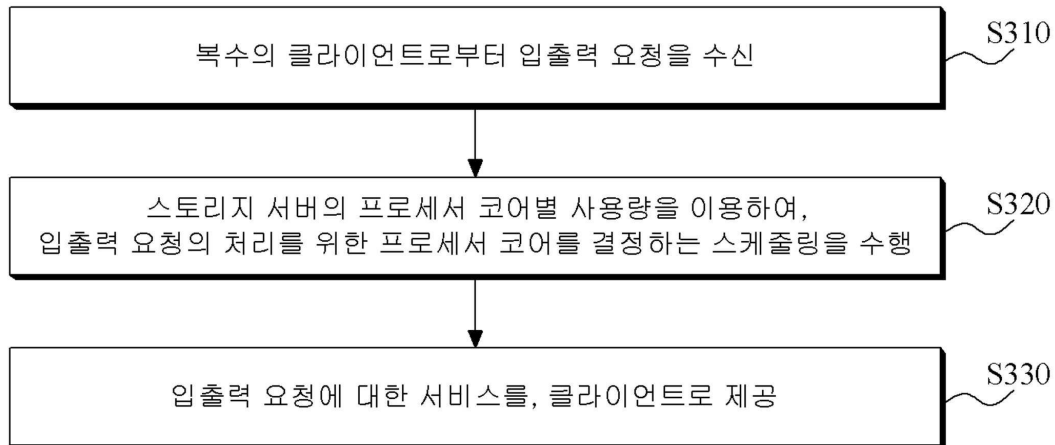
도면1



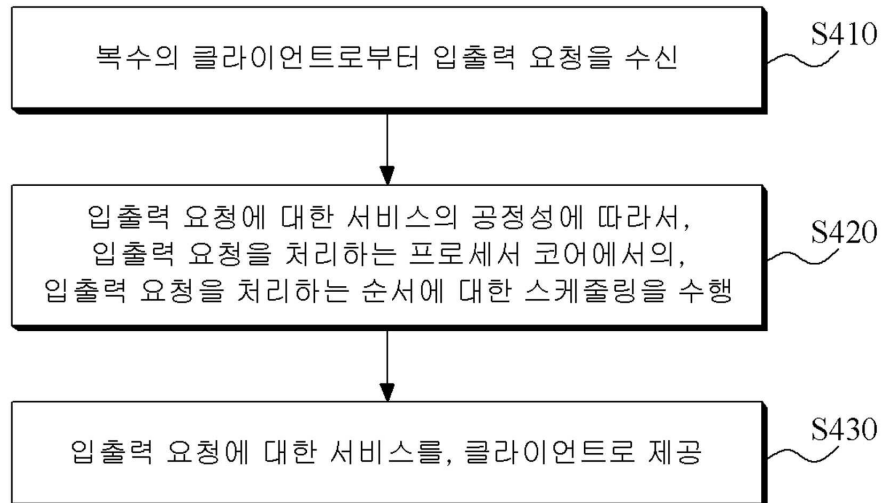
도면2



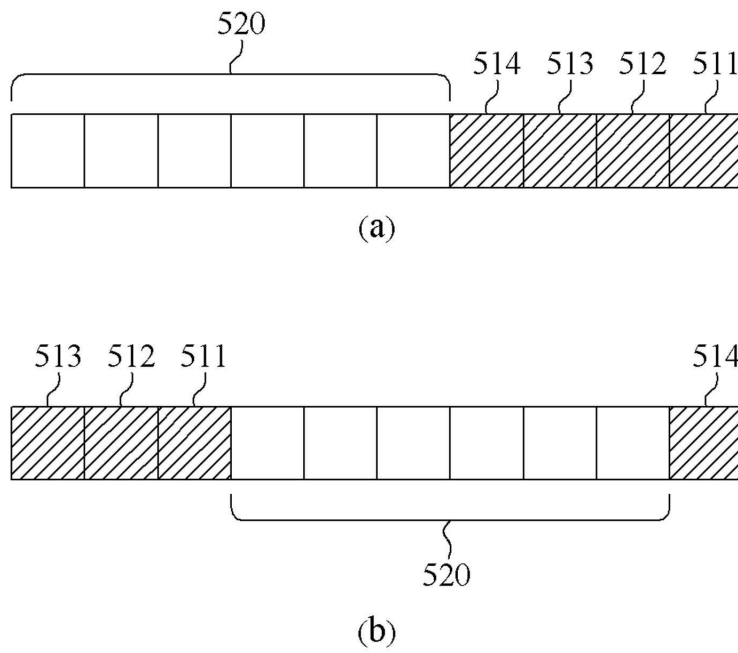
도면3



도면4



도면5



도면6

