



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2025년01월09일
(11) 등록번호 10-2750717
(24) 등록일자 2025년01월02일

(51) 국제특허분류(Int. Cl.)
H04N 21/2312 (2011.01) G06F 16/17 (2019.01)
G06F 16/75 (2019.01) G06N 3/047 (2023.01)
G06N 3/092 (2023.01) H04N 21/24 (2011.01)
(52) CPC특허분류
H04N 21/2312 (2013.01)
G06F 16/1737 (2019.01)
(21) 출원번호 10-2023-0164681
(22) 출원일자 2023년11월23일
심사청구일자 2023년11월23일
(56) 선행기술조사문헌
KR1020210132853 A*
US08316202 B1*
US20210240374 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
송민석
서울특별시 서초구 남부순환로 2311-12, 107동 402호 (방배동, 래미안방배아트힐)
권민구
경기도 수원시 영통구 봉영로1517번길 73, 921동 703호 (영통동, 삼성.태영 아파트)
(74) 대리인
양성보

전체 청구항 수 : 총 9 항

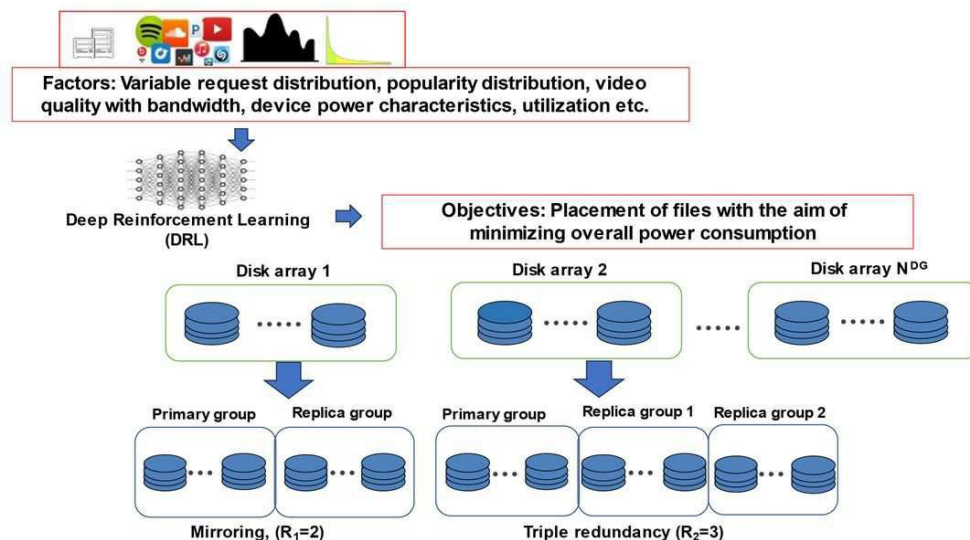
심사관 : 우정훈

(54) 발명의 명칭 디스크 배열 기반 스토리지 시스템에서 전력 효율 향상을 위한 비디오 파일 배치 기법

(57) 요약

디스크 배열 기반 스토리지 시스템에서 전력 효율 향상을 위한 비디오 파일 배치 기법이 개시된다. 일 실시예에 따른 파일 배치 시스템에 의해 수행되는 비디오 파일 배치 방법은, 비디오 스토리지 시스템을 위한 심층 강화 학습(Deep Reinforcement Learning; DRL) 기반의 비디오 파일 배치를 수행하는 단계를 포함하고, 상기 비디오 파일 배치를 수행하는 단계는, 가변적인 워크로드를 가진 복제 기반 디스크 어레이를 모델링하여 비디오 스토리지 시스템의 에너지 소비가 최소화되는 각 파일에 대한 디스크 어레이 인덱스의 값을 결정하는 단계를 포함할 수 있다.

대표도 - 도3



(52) CPC특허분류

G06F 16/75 (2019.01)

G06N 3/047 (2023.01)

G06N 3/092 (2023.01)

H04N 21/2405 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711198227
과제번호	RS-2023-00261534
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송기술국제공동연구
연구과제명	강화학습을 활용한 대규모 비디오 스트리밍 시스템의 에너지 최적화 기술 개발
기 여 율	1/1
과제수행기관명	인하대학교 산학협력단
연구기간	2023.07.01 ~ 2023.12.31

명세서

청구범위

청구항 1

파일 배치 시스템에 의해 수행되는 비디오 파일 배치 방법에 있어서,

비디오 스토리지 시스템을 위한 심층 강화 학습(Deep Reinforcement Learning; DRL) 기반의 비디오 파일 배치를 수행하는 단계

를 포함하고,

상기 비디오 파일 배치를 수행하는 단계는,

가변적인 워크로드를 가진 복제 기반 디스크 어레이를 모델링하여 비디오 스토리지 시스템의 에너지 소비가 최소화되는 각 파일에 대한 디스크 어레이 인덱스의 값을 결정하며, 비디오 스토리지 시스템의 에너지 소비를 최소화하기 위해 대역폭 제한 제약, 스토리지 제한 제약 및 디스크 어레이 할당 제약을 포함하는 제약 조건을 고려하여 각 파일에 대한 디스크 어레이 인덱스의 값을 결정하는 파일 배치 문제를 설정하고, 근위 정책 최적화 알고리즘을 통해 디스크 어레이 그룹에 대한 최대 유효 전력, 디스크 대역폭 및 저장 공간에 기초하여 각 파일의 배치를 결정하는 행동 공간, 현재 I/O 대역폭 활용을 기반으로 에이전트가 전력 효율적인 배치를 학습하는 관찰 공간, 현재 관찰된 상태에서 행동을 취하는 전환에 대해 생성되는 스칼라 값인 보상 함수로 구성된 마르코프 의사결정 프로세스로 모델링하는 단계

를 포함하는 비디오 파일 배치 방법.

청구항 2

제1항에 있어서,

상기 비디오 스토리지 시스템은, 복수 개의 이기종 스토리지 장치를 포함하는 디스크 어레이 그룹으로 구성되고,

상기 디스크 어레이 그룹은, 기본 그룹과 복수 개의 복제 그룹으로 구분되는 것이고,

상기 비디오 파일 배치를 수행하는 단계는,

체인형 디클러스터링을 통해 각 디스크 어레이의 복제 배치를 수행하는 단계

를 포함하는 비디오 파일 배치 방법.

청구항 3

삭제

청구항 4

제1항에 있어서,

상기 비디오 파일 배치를 수행하는 단계는,

근위 정책 최적화(PP0) 알고리즘을 사용하여 에이전트가 복수 개의 에피소드를 통해 가장 높은 보상을 받을 수 있는 정책을 학습하는 훈련 단계; 및

상기 학습을 통해 설정된 학습 모델에서 의사결정 프로세스를 통해 각 파일의 디스크 어레이 인덱스를 결정하는 결정 단계

를 포함하는 비디오 파일 배치 방법.

청구항 5

제4항에 있어서,

상기 에이전트는, 환경의 현재 상태에 대한 관찰을 수신한 다음 상기 관찰을 기반으로 행동을 취하고, 상기 행동에 대한 보상을 받고 환경을 새로운 상태로 전환하는 것

을 포함하는 비디오 파일 배치 방법.

청구항 6

제1항에 있어서,

상기 비디오 파일 배치를 수행하는 단계는,

각 파일 대한 행동이 선택됨에 따라 하나의 에피소드가 구성되고, 관찰된 파일의 상태 값을 기반으로 행동을 결정하고, 상기 결정된 행동의 결과로 보상 값을 제공하고, 파일의 상태 값을 업데이트하는 단계

를 포함하는 비디오 파일 배치 방법.

청구항 7

제6항에 있어서,

상기 관찰된 파일의 상태 값은, 기간에 대한 디스크 어레이 그룹의 대역폭 사용량, 기간에 대해 디스크 어레이 그룹이 소비하는 전력, 디스크 어레이 그룹의 스토리지 사용률, 모든 디스크 어레이 그룹에 대한 정규화된 유효 전력, 모든 디스크 어레이 그룹에 대한 정규화된 유틸 전력, 모든 디스크 어레이 그룹에 대한 정규화된 용량 제한, 모든 디스크 어레이 그룹에 대한 정규화된 대역폭 제한, 파일과 관련된 상태 값을 계산하고, 각 디스크 어레이 그룹의 대역폭, 스토리지 사용률, 전력 소비량이 업데이트되고, 각 파일의 스토리지 용량과 대역폭 수요의 정규화된 값으로 업데이트되는 것을 특징으로 하는 비디오 파일 배치 방법.

청구항 8

제6항에 있어서,

상기 보상값은, 행동의 선택에 따른 전력 소비량 증가를 기준으로 계산되는 것을 특징으로 하는 비디오 파일 배치 방법.

청구항 9

제1항에 있어서,

상기 비디오 파일 배치를 수행하는 단계는,

환경에서 수집된 관찰, 보상 및 작업의 시퀀스를 포함하는 궤적에 대해 확률적 경사 하강법을 사용하여 정책을 업데이트하고, 상기 업데이트된 정책을 통해 상태를 기반으로 작업을 결정하는 단계

를 포함하는 비디오 파일 배치 방법.

청구항 10

파일 배치 시스템에 있어서,

비디오 스토리지 시스템을 위한 심층 강화 학습(Deep Reinforcement Learning; DRL) 기반의 비디오 파일 배치를 수행하는 파일 배치 수행부

를 포함하고,

상기 파일 배치 수행부는,

가변적인 워크로드를 가진 복제 기반 디스크 어레이를 모델링하여 비디오 스토리지 시스템의 에너지 소비가 최소화되는 각 파일에 대한 디스크 어레이 인덱스의 값을 결정하며, 비디오 스토리지 시스템의 에너지 소비를 최소화하기 위해 대역폭 제한 제약, 스토리지 제한 제약 및 디스크 어레이 할당 제약을 포함하는 제약 조건을 고려하여 각 파일에 대한 디스크 어레이 인덱스의 값을 결정하는 파일 배치 문제를 설정하고, 근위 정책 최적화 알고리즘을 통해 디스크 어레이 그룹에 대한 최대 유효 전력, 디스크 대역폭 및 저장 공간에 기초하여 각 파일

의 배치를 결정하는 행동 공간, 현재 I/O 대역폭 활용을 기반으로 에이전트가 전력 효율적인 배치를 학습하는 관찰 공간, 현재 관찰된 상태에서 행동을 취하는 전환에 대해 생성되는 스칼라 값인 보상 함수로 구성된 마르코프 의사결정 프로세스로 모델링하는

것을 특징으로 하는 파일 배치 시스템.

발명의 설명

기술 분야

[0001] 아래의 설명은 비디오 파일 배치 기술에 관한 것이다.

배경 기술

[0003] 서버 시스템에서 데이터 배치 및 마이그레이션 기법을 사용하여 많은 스토리지 전력 관리 기법이 연구되었다. 예를 들면, I/O 대기열 크기와 전력 소비 사이의 관계를 조사하고 I/O 대기열 크기를 적응적으로 조정하여 전력 소비를 제한하는 방법이 제시되었다. 또한, 애플리케이션의 응답 시간을 고려하여 용량 제약이 있는 하드디스크 드라이브(HDD)를 일부 해제하는 파일 마이그레이션 방법이 제안되었다. 또한, 데이터베이스 스토리지 시스템에서 전력 소비와 쿼리 응답 시간 간의 균형을 맞추기 위해 디스크 상태를 동적으로 결정하고 디스크 간에 파일을 마이그레이션하는 기법이 제시되었다. 일부 기술은 콜드 스토리지 장치를 선택적으로 꺼서 디스크 에너지를 절약하기 위해 개발되었다. 예를 들면, 중복제거로 인한 콜드 스토리지 액세스 횟수 증가로 인한 읽기 성능 저하 문제를 제기하고, 허용 가능한 읽기 성능을 제공하면서 중복제거율을 개선할 수 있는 방법이 제안된 바 있으며, 온라인 메신저 분산 스토리지 시스템에서 사용할 수 있는 콜드 HDD 장치의 수를 점진적으로 증가시키는 기법을 구현한 결과가 제시된 바 있다.

[0004] 다른 예로서, 모바일 메신저 애플리케이션을 위한 콜드 스토리지 시스템 벤치마킹 툴을 소개하고 이를 기반으로 콜드 스토리지 테스트베드를 구축한 결과가 제시되었으며, 일반적인 스토리지 스택을 자동으로 파라미터화하는 nearline HDD 기반 아카이브 스토리지 시스템이 제시되었다. 그러나 이러한 연구들은 모두 일반 데이터 저장을 위해 설계되었으며, 영상 파일의 특성을 고려하지 않았기 때문에 영상 저장 시스템에 활용하기 어려울 수 있다. 이에 다중 속도 디스크를 사용하는 비디오 스토리지 시스템을 위해 HDD의 회전 속도를 최소화하여 전력 소비를 감소시키는 SSD 관리 기법이 제안되었다.

[0005] 그러나, 가변적인 워크로드 환경에서 이기종 디바이스의 전력 특징을 고려한 기존 연구는 없는 것으로 파악되었다.

발명의 내용

해결하려는 과제

[0007] 본 발명은 Netflix 및 비디오 스트리밍 서비스와 같은 대규모 비디오 스토리지 시스템에서 발생하는 높은 에너지 소비 문제를 효과적으로 해결하는 방법이다. 이기종 장치의 전력 특성, 가변 워크로드, 디스크 어레이 모델 등의 다양한 변수를 고려하여 스토리지 에너지 소비를 최소화하는 심층 강화 학습을 활용한 새로운 파일 배치 알고리즘을 제안한다.

[0008] 대규모 비디오 스토리지 시스템에서 파일 배치를 최적화하여 에너지 소비를 최소화하는 것을 목표로 하고, 이를 위해 이기종 장치의 전력 특성, 가변 워크로드, 디스크 어레이 모델 등의 다양한 변수를 고려해 심층 강화 학습을 활용하여 효과적인 알고리즘을 개발해 적용한다.

과제의 해결 수단

[0010] 파일 배치 시스템에 의해 수행되는 비디오 파일 배치 방법은, 비디오 스토리지 시스템을 위한 심층 강화 학습(Deep Reinforcement Learning; DRL) 기반의 비디오 파일 배치를 수행하는 단계를 포함하고, 상기 비디오 파일 배치를 수행하는 단계는, 가변적인 워크로드를 가진 복제 기반 디스크 어레이를 모델링하여 비디오 스토리지 시스템의 에너지 소비가 최소화되는 각 파일에 대한 디스크 어레이 인덱스의 값을 결정하는 단계를 포함할 수 있다.

[0011] 상기 비디오 스토리지 시스템은, 복수 개의 이기종 스토리지 장치를 포함하는 디스크 어레이 그룹으로

구성되고, 상기 디스크 어레이 그룹은, 기본 그룹과 복수 개의 복제 그룹으로 구분되는 것이고, 상기 비디오 파일 배치를 수행하는 단계는, 체인형 디클러스터링을 통해 각 디스크 어레이의 복제 배치를 수행하는 단계를 포함할 수 있다.

[0012] 상기 비디오 파일 배치를 수행하는 단계는, 비디오 스토리지 시스템의 에너지 소비를 최소화하기 위해 대역폭 제한 제약, 스토리지 제한 제약 및 디스크 어레이 할당 제약을 포함하는 제약 조건을 고려하여 각 파일에 대한 디스크 어레이 인덱스의 값을 결정하는 파일 배치 문제를 설정하는 단계를 포함할 수 있다.

[0013] 상기 비디오 파일 배치를 수행하는 단계는, 근위 정책 최적화(PPO) 알고리즘을 사용하여 에이전트가 복수 개의 에피소드를 통해 가장 높은 보상을 받을 수 있는 정책을 학습하는 훈련 단계; 및 상기 학습을 통해 설정된 학습 모델에서 의사결정 프로세스를 통해 각 파일의 디스크 어레이 인덱스를 결정하는 결정 단계를 포함할 수 있다.

[0014] 상기 에이전트는, 환경의 현재 상태에 대한 관찰을 수신한 다음 상기 관찰을 기반으로 행동을 취하고, 상기 행동에 대한 보상을 받고 환경을 새로운 상태로 전환하는 것이고, 상기 비디오 파일 배치를 수행하는 단계는, 상기 근위 정책 최적화 알고리즘을 통해 디스크 어레이 그룹에 대한 최대 유효 전력, 디스크 대역폭 및 저장 공간에 기초하여 각 파일의 배치를 결정하는 행동 공간, 현재 I/O 대역폭 활용을 기반으로 에이전트가 전력 효율적인 배치를 학습하는 관찰 공간, 현재 관찰된 상태에서 행동을 취하는 전환에 대해 생성되는 스칼라 값인 보상 함수로 구성된 마르코프 의사결정 프로세스로 모델링하는 단계를 포함할 수 있다.

[0015] 상기 비디오 파일 배치를 수행하는 단계는, 각 파일 대한 행동이 선택됨에 따라 하나의 에피소드가 구성되고, 관찰된 파일의 상태 값을 기반으로 행동을 결정하고, 상기 결정된 행동의 결과로 보상 값을 제공하고, 파일의 상태 값을 업데이트하는 단계를 포함할 수 있다.

[0016] 상기 관찰된 파일의 상태 값은, 기간에 대한 디스크 어레이 그룹의 대역폭 사용량, 기간에 대해 디스크 어레이 그룹이 소비하는 전력, 디스크 어레이 그룹의 스토리지 사용률, 모든 디스크 어레이 그룹에 대한 정규화된 유효 전력, 모든 디스크 어레이 그룹에 대한 정규화된 유휴 전력, 모든 디스크 어레이 그룹에 대한 정규화된 용량 제한, 모든 디스크 어레이 그룹에 대한 정규화된 대역폭 제한, 파일과 관련된 상태 값을 계산하고, 각 디스크 어레이 그룹의 대역폭, 스토리지 사용률, 전력 소비량이 업데이트하고, 각 파일의 스토리지 용량과 대역폭 수요의 정규화된 값으로 업데이트할 수 있다.

[0017] 상기 보상값은, 행동의 선택에 따른 전력 소비량 증가를 기준으로 계산되는 것일 수 있다.

[0018] 상기 비디오 파일 배치를 수행하는 단계는, 환경에서 수집된 관찰, 보상 및 작업의 시퀀스를 포함하는 궤적에 대해 확률적 경사 하강법을 사용하여 정책을 업데이트하고, 상기 업데이트된 정책을 통해 상태를 기반으로 작업을 결정하는 단계를 포함할 수 있다.

[0019] 파일 배치 시스템은, 비디오 스토리지 시스템을 위한 심층 강화 학습(Deep Reinforcement Learning; DRL) 기반의 비디오 파일 배치를 수행하는 파일 배치 수행부를 포함하고, 상기 파일 배치 수행부는, 가변적인 워크로드를 가진 복제 기반 디스크 어레이를 모델링하여 비디오 스토리지 시스템의 에너지 소비가 최소화되는 각 파일에 대한 디스크 어레이 인덱스의 값을 결정할 수 있다.

발명의 효과

[0021] 심층 강화 학습 기반의 파일 배치 알고리즘을 제안함으로써 스토리지 에너지 소비를 감소시킬 수 있다.

도면의 간단한 설명

[0023] 도 1은 일 실시예에 있어서, 파일 배치 시스템의 구성을 설명하기 위한 블록도이다.

도 2는 일 실시예에 있어서, 비디오 파일 배치 방법을 설명하기 위한 흐름도이다.

도 3은 일 실시예에 있어서, 비디오 시스템의 아키텍처 및 모델링 동작을 설명하기 위한 도면이다.

도 4는 일 실시예에 있어서, 각 디스크 어레이의 복제 배치에 체인형 디클러스터링을 수행하는 동작을 설명하기 위한 도면이다.

도 5는 일 실시예에 있어서, PPO의 기본 하이퍼 파라미터를 설명하기 위한 도면이다.

도 6 및 도 7은 일 실시예에 있어서, 훈련 단계와 결정 단계로 구성된 FPA의 의사코드를 설명하기 위한 예이다.

도 8은 일 실시예에 있어서, 10개의 상용 SSD의 전력, 대역폭 및 저장 용량의 특성을 표로 나타낸 도면이다.

도 9는 일 실시예에 있어서, 미러링 복제 환경에서 적용 시 타 기법 대비 제안 기법의 전력 평균 감소율을 나타낸 그래프이다.

도 10은 일 실시예에 있어서, 미러링 복제 환경에서 타 기법 대비 제안 기법의 전력 최대 감소율 나타낸 그래프이다.

도 11은 일 실시예에 있어서, 미러링과 트리플 중복(triple redundancy)을 혼합 사용할 시 타 기법 대비 제안 기법의 전력 평균 감소율을 나타낸 그래프이다.

도 12는 일 실시예에 있어서, 미러링과 트리플 중복을 혼합 사용할 시 타 기법 대비 제안 기법의 최대 감소율을 나타낸 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0024] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0026] 비디오 스트리밍 서비스가 인기를 끌면서 비디오 스토리지 시스템의 높은 에너지 소비 문제를 해결하는 것이 중요해졌다. 그러나 이기종 장치의 전력 특성, 가변 워크로드, 디스크 어레이(array) 모델 등 다양한 요인들이 스토리지 에너지 소비에 큰 영향을 미치기 때문에 결정론적인 정책으로 에너지 소비를 최소화하는 것은 효과적이지 않다. 실시예에서는 복제 기반 비디오 스토리지 서버를 위한 심층 강화 학습(Deep Reinforcement Learning; DRL) 기반의 새로운 파일 배치 동작에 대하여 설명하기로 하며, 이는, 전체 스토리지 에너지 소비를 최소화하는 것을 목표로 한다. 먼저, 시간에 따라 변하는 스트리밍 워크로드를 가진 복제 기반 디스크 어레이를 심층 강화 학습 환경으로 모델링하여 에이전트가 전력 효율적인 파일 배치를 찾는 것을 목표로 설정한 후, 각 파일의 배치를 결정하는 행동 공간, 현재 I/O 대역폭 활용을 기반으로 에이전트가 전력 효율적인 배치를 학습할 수 있는 관찰 공간(Observation space), 각 행동에 대해 더 높은 전력 소비에 대해 더 높은 페널티를 제공하는 보상 모델, 에이전트가 불필요한 행동을 선택하지 않도록 하여 효과적인 학습을 지원하는 액션 마스킹 모델로 구성된 근위 정책 최적화(Proximal Policy Optimization; PPO) 알고리즘을 제안한다. 다양한 저장 장치 모델, 복제 방법, 비디오 품질 시나리오에서 제안된 방식을 평가하기 위해 광범위한 시뮬레이션이 수행되었고, 그 결과로 당사의 계획이 파일 배치에 효과적인 것으로 알려진 기존 벤치마크 계획에 비해 스토리지 에너지 소비를 2.49%~27.38%(평균 11.7%) 감소시키는 것으로 나타났다

[0027] 도 1은 일 실시예에 있어서, 파일 배치 시스템의 구성을 설명하기 위한 블록도이고, 도 2는 일 실시예에 있어서, 비디오 파일 배치 방법을 설명하기 위한 흐름도이다.

[0028] 파일 배치 시스템(100)의 프로세서는 파일 배치 수행부(110)를 포함할 수 있다. 이러한 프로세서의 구성요소는 파일 배치 시스템에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 프로세서 및 프로세서의 구성요소는 도 2의 비디오 파일 배치 방법이 포함하는 단계(210)를 수행하는 파일 배치 시스템을 제어할 수 있다. 이때, 프로세서 및 프로세서의 구성요소는 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다.

[0029] 프로세서는 비디오 파일 배치 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩할 수 있다. 예를 들면, 파일 배치 시스템에서 프로그램이 실행되면, 프로세서는 운영체제의 제어에 따라 프로그램의 파일로부터 프로그램 코드를 메모리에 로딩하도록 파일 배치 시스템을 제어할 수 있다. 이때 프로세서는 파일 배치 수행부(110)에 메모리에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계(210)를 실행하기 위한 프로세서의 서로 다른 기능적 표현들일 수 있다.

[0030] 단계(210)에서 파일 배치 수행부(110)는 비디오 스토리지 시스템을 위한 심층 강화 학습(Deep Reinforcement Learning; DRL) 기반의 비디오 파일 배치를 수행할 수 있다. 보다 상세하게는, 파일 배치 수행부(110)는 가변적인 워크로드를 가진 복제 기반 디스크 어레이를 모델링하여 비디오 스토리지 시스템의 에너지 소비가 최소화되는 각 파일에 대한 디스크 어레이 인덱스의 값을 결정할 수 있다. 파일 배치 수행부(110)는 체인형 디클러스터링을 통해 각 디스크 어레이의 복제 배치를 수행할 수 있다. 파일 배치 수행부(110)는 비디오 스토리지 시스템의 에너지 소비를 최소화하기 위해 대역폭 제한 제약, 스토리지 제한 제약 및 디스크 어레이 할당 제약을 포함하는 제약 조건을 고려하여 각 파일에 대한 디스크 어레이 인덱스의 값을 결정하는 파일 배치 문제를 설정할 수 있다. 파일 배치 수행부(110)는 근위 정책 최적화(PPO) 알고리즘을 사용하여 에이전트가 복수 개의 에피소

드를 통해 가장 높은 보상을 받을 수 있는 정책을 학습할 수 있다. 파일 배치 수행부(110)는 학습을 통해 설정된 학습 모델에서 의사결정 프로세스를 통해 각 파일의 디스크 어레이 인덱스를 결정할 수 있다. 파일 배치 수행부(110)는 각 파일 대한 행동이 선택됨에 따라 하나의 에피소드가 구성되고, 관찰된 파일의 상태 값을 기반으로 행동을 결정하고, 결정된 행동의 결과로 보상 값을 제공하고, 파일의 상태 값을 업데이트할 수 있다. 파일 배치 수행부(110)는 환경에서 수집된 관찰, 보상 및 작업의 시퀀스를 포함하는 궤적에 대해 확률적 경사 하강법을 사용하여 정책을 업데이트하고, 업데이트된 정책을 통해 상태를 기반으로 작업을 결정할 수 있다.

[0031] 도 3은 일 실시예에 있어서, 비디오 시스템의 아키텍처 및 모델링 동작을 설명하기 위한 도면이다.

[0032] 도 3에서는 비디오(VoD) 스토리지 시스템을 예를 들어 설명하기로 한다. 이기종 스토리지 장치의 N^{DG} 디스크 어레이 그룹(Disk array Group; DG)으로 구성되어 있으며, 각 DG의 스토리지 장치는 기본 그룹과 여러 개의 복제 그룹으로 나눌 수 있다. R_i 를 디스크 어레이 i 의 복사본 수인 복제 인자라고 한다. 예를 들어, $R_i = 2$ 인 경우, 디스크 어레이 i 에는 단일 복제(redundancy)를 사용한 미러링이 사용된다. 각 디스크 어레이 i 는 용량, 대역폭 및 전력 특성이 동일한 동종 디스크 N_i^{disk} 로 구성되어 있다고 가정한다. 비디오 파일 배치는 VoD 스토리지 시스템의 에너지 소비와 처리량에 상당한 영향을 미친다. 예를 들어, 워크로드가 디스크 어레이에 적절하게 분배되지 않으면 로드 불균형으로 인해 처리량이 감소할 수 있다. 마찬가지로, 많은 요청이 고출력 저장 장치에서 처리되면 에너지 소비가 증가한다. 따라서, 예상되는 워크로드를 기반으로 파일을 효과적으로 배치하는 것이 중요하며, 일반적으로 VoD 스토리지 시스템에서 데이터 배치는 수요가 적은 시간대(예를 들면, 동트기 전)에 이루어진다. 높은 데이터 가용성을 제공하기 위해서는 VoD 스토리지 시스템에서 복제가 필수적이다. 도 4와 같이, 파일 배치 시스템은 각 디스크 어레이의 복제 배치에 체인형 디클러스터링을 사용할 수 있다. 이는 높은 데이터 가용성을 제공하고 워크로드 및 디스크 장애의 변화에 강력하게 대응하는 것으로 알려져 있기 때문이다.

[0033] B_i^{DG} 를 디스크 배열 i 의 대역폭 요구조건이라고 하자. 그러면 B_i^{DG} 는 $B_i^{DG} = B_i^{disk} N_i^{disk}$ 으로 계산된다. 비디오 스트리밍 시스템에서 사용자 요청의 분포는 일반적으로 시간별 분포를 따르며, 요청 수는 하루 종일 다양하다. 따라서, 디스크 어레이의 대역폭 사용률은 하루 종일 크게 달라질 수 있다. 이를 표현하기 위해 대역폭 요구량이 매 주기마다 변경되어 각 파일 j 의 대역폭 요구량이 k 번째 주기($k = 1, \dots, N^{period}$)에 걸쳐 $B_{j,k}$ 가 되는 VoD 스토리지 시스템을 고려한다. 여기서, N^{period} 은 주기 수이다. 실시예에서의 목적은 전체 스토리지 전력 소비를 최소화하기 위해 각 파일 j 에 대한 디스크 어레이 인덱스 $X_j(j = 1, \dots, N^{file})$ 를 구하는 것이다. 여기서, N 파일은 비디오 파일 수이다. $\vec{X} = X_1, \dots, X_{N^{file}}$ 인 경우, k 번째 기간 동안 디스크 어레이 i 의 대역폭 사용률은 다음과 같이 $\vec{X}$ 의 함수로 표현할 수 있다. $U_{i,k}(\vec{X})$. 그러면 디스크 어레이 i 에 대한 총 대역폭 수요를 B_i^{DG} 로 나누면 다음과 같이 $U_{i,k}(\vec{X})$ 를 계산할 수 있다.

[0034] 수학적 식 1:

$$U_{i,k}(\vec{X}) = \frac{\sum_{\forall j} X_j = i B_{j,k}}{B_i^{DG}},$$

[0035]

[0036] $U_{i,k} \leq 1/R_i$ 인 경우 복제 그룹에 있는 모든 디스크를 꺼서 에너지 소비를 크게 절감할 수 있다. 예를 들어, $U_{i,k} \leq 0.5$ 이고 $R_i = 2$ 이면 어레이에 있는 디스크의 절반을 끌 수 있다. $U_{i,k} \leq 1/3$ 이고 $R_i = 3$ 이면 어레이에 있는 디스크의 $2/3$, $1/3 \leq U_{i,k} \leq 2/3$ 이면 어레이에 있는 디스크의 $1/3$ 이 꺼질 수 있다. 이는 대역폭 사용률을 제어하여 에너지 소비를 크게 줄일 수 있는 여지가 있음을 나타낸다. 따라서, 디스크 어레이 i 에 대한 k 번째 기간 동안의 평균 전력 소비량인 $P_{i,k}(\vec{X})$ 는 $\vec{X}$ 의 함수로 표현된다. 각 디스크 어레이는 각기 다른 전력 특성을 가지며, 어레이 i 에 대한 활성 전력 P_i^A 와 유휴 전력 P_i^I 가 있다. 디스크 어레이 i 에 대한 k 번째 기간 동안의 유휴 전력 및 활성 전력 소비량을 각각 $P_{i,k}^I(\vec{X})$ 및 $P_{i,k}^A(\vec{X})$ 라고 한다. 그러면 $P_{i,k}^I(\vec{X})$ 는 다

음과 같이 계산될 수 있다.

수학식 2:

$$P_{i,k}^A(\vec{X}) = P_i^A U_{i,k}(\vec{X})$$

그러나, 유틸 전력은 현재 CPU 사용률과 복제 방식에 따라 다르게 계산될 수 있다. 설명의 편의를 위해 두 가지 복제 방법만 고려한다. $R_i = 2$ 및 $R_i = 3$. 그러면 $P_{i,k}^I$ 는 다음과 같이 표현할 수 있다.

● Mirroring case($R_i = 2$)

수학식 3:

$$P_{i,k}^I(\vec{X}) = \begin{cases} \frac{1}{2}(1 - U_{i,k}(\vec{X}))P_i^I & \text{if } U_{i,k}(\vec{X}) < \frac{1}{2} \\ (1 - U_{i,k}(\vec{X}))P_i^I & \text{otherwise.} \end{cases}$$

● Triple redundancy case($R_i = 3$)

수학식 4:

$$P_{i,k}^I(\vec{X}) = \begin{cases} \frac{1}{3}(1 - U_{i,k}(\vec{X}))P_i^I & \text{if } U_{i,k}(\vec{X}) < \frac{1}{3} \\ \frac{2}{3}(1 - U_{i,k}(\vec{X}))P_i^I & \text{if } \frac{1}{3} \leq U_{i,k}(\vec{X}) < \frac{2}{3} \\ (1 - U_{i,k}(\vec{X}))P_i^I & \text{otherwise.} \end{cases}$$

따라서, $P_{i,k}(\vec{X})$ 는 다음과 같이 계산된다.

수학식 5:

$$P_{i,k}(\vec{X}) = P_{i,k}^I(\vec{X}) + P_{i,k}^A(\vec{X})$$

디스크 어레이 i 가 제공하는 스토리지 용량(GB)과 대역폭(GB/s)을 각각 C_i^{DG} 와 B_i^{DG} 라고 기재하기로 한다. 각 파일 j 에는 k 번째 기간 동안 C_j^{file} 의 저장 공간과 B_j^{file} 의 대역폭이 필요하다. 그러면 파일 배치 문제(FPP)에는 다음과 같은 세 가지 제약 조건이 있다.

● 대역폭 제한 제약: 각 기간 k , ($k = 1, \dots, N^{period}$)에 대해 디스크 어레이 i , ($i = 1, \dots, N^{DG}$)의 모든 파일을 처리하는 데 필요한 대역폭 사용량은 다음과 같이 B_i^{DG} 를 초과하지 않아야 한다.

수학식 6:

$$\forall i \text{ and } k, \sum_{\forall j, X_j=i} B_{j,k}^{file} \leq B_i^{DG}$$

● 스토리지 제한 제약: 각 디스크 어레이에 저장된 총 파일 크기는 다음과 같이 스토리지 용량을 초과하지 않아야 한다.

수학식 7:

$$\forall i (i = 1, \dots, N^{DG}), \sum_{\forall j, X_j=i} C_j^{file} \leq C_i^{DG}$$

[0056] ● 디스크 어레이 할당 제약 조건: 각 파일은 디스크 어레이 중 하나에 저장되어야 하므로 $X_j = 1, \dots, N^{DG}$ 가 된다. 이를 바탕으로 전력 소비를 최소화하기 위해 각 파일 j에 대한 X_j 의 값을 결정하는 FPP를 다음과 같이 공식화한다.

[0057] 수학적식 8:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{N^{DG}} \sum_{k=1}^{N^{\text{period}}} P_{i,k}(\vec{X}), \\ \text{Subject to} \quad & X_j = 1, \dots, N^{DG}, \\ & \forall i \ (i = 1, \dots, N^{DG}), \sum_{j, X_j=i} C_j^{\text{file}} \leq C_i^{DG}, \\ & \forall i \text{ and } k, \sum_{j, X_j=i} B_{j,k}^{\text{file}} \leq B_i^{DG}. \end{aligned}$$

[0058]

[0059] 스토리지 전력 소비는 수학적식 2, 3, 4, 5에 설명된 대로 대역폭 사용률에 크게 영향을 받기 때문에 모든 기간 동안 디스크 대역폭 요구 사항의 변화를 반영하는 것이 필수적이며, 이를 위해서는 다음과 같은 측면에서 DRL과 같은 학습 기반 방법이 필요하다.

[0060] 첫째, 각 어레이 그룹의 대역폭 사용률은 다른 파일이 배치되는 방식에 따라 결정되므로 모든 파일 배치 조합을 고려해야 하며, 이는 계산적으로 다루기 어렵다.

[0061] 둘째, 대역폭 사용률은 주기마다 변하기 때문에 에너지 효율적인 데이터 배치는 주기마다 달라지므로 결정론적 알고리즘을 사용하기 어렵다. 이는 시뮬레이션을 통한 학습을 통한 최적화가 다양한 스트리밍 워크로드에 효과적이라는 것을 시사한다.

[0062] 파일 배치 시스템은 훈련 단계와 결정 단계로 구성될 수 있다. 트레이닝 단계에서는 PPO 알고리즘을 사용하여 에이전트가 가장 높은 보상을 받을 수 있는 정책을 학습한다. PPO는 매우 많은 수의 에피소드를 통해 학습하며, 모든 에피소드가 완료되면 학습 프로세스가 종료된다. 학습 단계 끝나면 학습 모델이 설정되며, 설정된 학습 모델은 의사 결정 단계에서 각 파일 j의 X_j 를 결정하는 데 사용할 수 있다. 에이전트와 환경의 상호 작용은 일련의 스텝으로 구성된 에피소드로 구성된다. 각 스텝에서 에이전트는 환경의 현재 상태에 대한 관찰을 수신한 다음 해당 관찰을 기반으로 행동을 취한다. 그런 다음, 에이전트는 해당 행동에 대한 보상을 받고 환경을 새로운 상태로 전환한다. 이러한 상호작용을 효과적으로 표현하기 위해 PPO의 의사 결정 프로세스는 행동 공간, 관찰 공간, 현재 관찰된 상태에서 행동을 취하는 전환에 대해 생성되는 스칼라 값인 보상 함수로 구성된 마르코프 의사 결정 프로세스로 모델링할 수 있다. 그런 다음 환경은 초기화, 재설정, 스텝 및 상태 업데이트를 포함하는 네 가지 모듈로 구성된다.

[0063] 초기화 모듈은 PPO 알고리즘이 시작될 때 실행된다. 초기화 모듈은 행동 및 관찰 공간에 필요한 메모리 공간을 할당한다.

[0064] 재설정 모듈은 새로운 에피소드가 시작되면 행동 및 관찰 값 등 각 에피소드를 시작하는 데 필요한 데이터 구조를 초기화한다.

[0065] 스텝 모듈은 관찰을 기반으로 행동을 취한 후 행동의 결과로 보상을 받는다.

[0066] 상태 업데이트 모듈은 스텝 모듈이 끝나면 관찰 공간의 상태 값이 업데이트된다.

[0067] 각 에피소드는 다음 모듈에 대한 연속적인 호출로 구성된다. 리셋 모듈이 먼저 호출된 후 스텝 및 상태 업데이트의 두 모듈에 대한 N^{file} 호출이 이어진다.

[0068] PPO는 정책을 업데이트하고 현재 관찰(관측)에 따라 행동을 결정한다. 모든 기간에 걸쳐 전력 소비를 최소화하기 위해 각 기간($k = 1, \dots, N^{\text{period}}$)의 각 DG에 대한 전력 소비, 디스크 대역폭 및 저장 공간 활용도가 관찰 공간에 포함된다. 따라서, 각 스텝에서 이러한 상태 값은 작업 선택의 결과로 업데이트된다. PPO 알고리즘의 정확도를 높이기 위해 일반적으로 관찰 값을 정규화 하므로 각 DG의 특성을 나타내는 관찰 값(유효 및 유휴 전력, 저장 용량 및 대역폭)은 최대값을 기준으로 정규화 된다. 이를 표현하기 위해 P^{max} , C^{max} 및 B^{max} 를 각각 모든 DGs i , ($i = 1, \dots, N^{DG}$)에 대한 최대 유효 전력, 저장 공간 및 대역폭이라고 한다. 행동 공간 A 는 $A = \{1, \dots,$

N^{DG} 로 표현되며, 각 스텝 j 에서 각 파일 j 에 대해 행동 $a_j \in \{1, \dots, N^{DG}\}$ 가 선택되므로 하나의 에피소드가 N^{file} 스텝으로 구성됨을 나타낸다. 각 스텝에서 PPO는 관찰된 상태 값 집합인 S_j 를 기반으로 행동 a_j 를 결정한다. j 번째 스텝이 시작될 때 파일 j 의 상태 값이 업데이트된다. 그런 다음, S_j 는 벡터 $\vec{A}_j = \{a_1, \dots, a_j\}$ 의 함수와 다음 하위 집합의 합으로 표현할 수 있다.

[0069] 1) $S_j^{bw_util} = \{U_{1,1}(\vec{A}_j), \dots, P_{NDG, N^{period}}(\vec{A}_j)\}$, 여기서 $U_{i,k}(\vec{A}_j)$ 는 반복 j 후 기간 k , ($k = 1, \dots, N^{period}$)에 대한 DG i , ($i = 1, \dots, N^{DG}$)의 대역폭 사용량을 나타낸다.

[0070] 2) $S_j^{qwr} = \{P_{1,k}(\vec{A}_j), \dots, P_{NDG, N^{period}}(\vec{A}_j)\}$, 여기서 $P_{i,k}(\vec{A}_j)$ 는 반복 j 후 기간 k , ($k = 1, \dots, N^{period}$)에 대해 DG i , ($i = 1, \dots, N^{DG}$)가 소비하는 전력을 나타내며, 수학적 식 5를 사용하여 계산할 수 있다.

[0071] 3) $S_j^{st_util} = \{U_1^{st}(\vec{A}_j), \dots, U_{N^{DG}}^{st}(\vec{A}_j)\}$, 여기서 $U_i^{st}(\vec{A}_j)$ 는 DG i , ($i = 1, \dots, N^{DG}$)의 스토리지 사용률

을 나타내며, 다음과 같이 계산할 수 있다: $U_i^{st}(\vec{A}_j) = \frac{\sum_{j=1}^N x_{j=i} S_j^{file}}{S_1^{DG}}$

[0072] 4) 모든 DG에 대한 정규화된 유효 전력의 배열인 S_j^A 는 다음과 같이 계산된다: $\{\frac{P_1^A}{p^{max}}, \dots, \frac{P_{NDG}^A}{p^{max}}\}$.

[0073] 5) 모든 DG에 대한 정규화된 유효 전력의 배열인 S_j^I 는 다음과 같이 계산된다: $\{\frac{P_1^I}{p^{max}}, \dots, \frac{P_{NDG}^I}{p^{max}}\}$.

[0074] 6) 모든 DG에 대한 정규화된 용량 제한의 배열인 S_j^C 는 다음과 같이 계산된다: $\{\frac{C_1^{DG}}{C^{max}}, \dots, \frac{C_{NDG}^{DG}}{C^{max}}\}$.

[0075] 7) 모든 DG에 대한 정규화된 대역폭 제한의 배열인 S_j^B 는 다음과 같이 계산된다: $\{\frac{B_1^{DG}}{B^{max}}, \dots, \frac{B_{NDG}^{DG}}{B^{max}}\}$.

[0076] 8) 파일 j 와 관련된 상태 값의 배열인 S_j^{file} 는 다음과 같이 도출된다: $\{\frac{C_j^{file}}{C^{max}}, \frac{B_{j,1}^{file}}{B^{max}}, \dots, \frac{B_{j,N^{period}}^{file}}{B^{max}}\}$.

[0077] S_j 를 구성하는 하위 집합은 다음 세 가지 클래스로 분류된다.

[0078] 1) $S_j^A, S_j^I, S_j^C, S_j^B$: 각 DG의 특성을 표현하는 상수 상태 값을 포함하는 하위 집합이다.

[0079] 2) $S_j^{bw_util}, S_j^{pwr}, S_j^{st_util}$: 각 스텝 j 에서 각 파일 j 에 대한 동작 값 X_j 가 결정된다. 파일 j 가 DG A_j 에 저장되어 있다고 가정하면 각 DG의 대역폭, 스토리지 사용률, 전력 소비량이 업데이트된다.

[0080] 3) S_j^{file} : 각 스텝 j 에서 이 하위 집합은 각 파일 j 의 스토리지 용량과 대역폭 수요의 정규화된 값으로 업데이트된다.

[0081] DG마다 활성 및 유효 전력 특성이 다르고, 현재 대역폭 사용률에 따라 전체 전력이 달라지므로 다른 행동을 선택하면 전력 소비량도 달라진다. 따라서, 보상 값은 행동 선택에 따른 전력 소비량 증가를 기준으로 계산된다. 이를 음수로 설정하면 전력 소비 증가율이 낮은 행동이 더 높은 보상 값을 갖게 된다. $step_j$ 에서 각 행동 a_j

가 선택될 때마다 수학적 식 8과 같이 다음과 같은 보상 $r(S_j, a_j)$ 이 계산된다.

[0082] 수학적 식 8:

$$r(S_j, a_j) = \begin{cases} -\frac{\sum_{k=1}^{N^{\text{period}}} P_{a_j, k}(\bar{A}_j) - P_{a_j, k}(\bar{A}_{j-1})}{N^{\text{period}}} & j = 2, \dots, N^{\text{file}} \\ -\frac{\sum_{k=1}^{N^{\text{period}}} P_{a_j, k}(\bar{A}_j)}{N^{\text{period}}} & j = 1. \end{cases}$$

[0084] $r(S_j, a_j)$ 는 행동 a_j 후 전력 소비 증가의 음수를 나타내므로 전력 소비가 많이 증가할수록 보상은 낮아진다.

[0085] 훈련 단계에서 PPO 알고리즘은 환경에서 궤적(관찰, 보상 및 작업의 시퀀스)을 수집하고 잘린 대리 목적 함수를 계산하며 확률적 경사 하강법을 사용하여 정책을 업데이트한다. 그런 다음, 업데이트된 정책을 사용하여 관찰된 상태를 기반으로 작업을 결정한다. PPO를 포함한 정책 그라데이션 방법은 매개변수화된 정책 π_{θ_t} 를 직접 최적화하여 θ_t 의 매개변수가 각 시간 스텝 t 에서 조정되어 주어진 측정값을 달성한다. PPO는 주어진 측정값에 대해 잘린 대리 목적을 고려하고 다음 매개변수 θ_{t+1} 을 다음과 같이 업데이트한다.

[0086] 수학적 식 9:

$$\theta_{t+1} = \arg \max_{\theta} E_{\pi \sim \pi_{\theta_t}} [L_{\theta_t}(\theta)]$$

[0088] 여기서 $L_{\theta_t}(\theta)$ 는

[0089] 수학적 식 10:

$$L_{\theta_t}(\theta) = \min\left(\frac{\pi_{\theta}}{\pi_{\theta_t}} A^{\pi_{\theta_t}}, \text{clip}\left(\frac{\pi_{\theta}}{\pi_{\theta_t}}, 1 - \epsilon, 1 + \epsilon\right) A^{\pi_{\theta_t}}\right)$$

[0091] 여기서, ϵ 는 대규모 정책 업데이트를 방지하기 위해 정책 업데이트를 $\frac{\pi_{\theta}}{\pi_{\theta_t}}$ 로 자르는 데 사용되는 하이퍼 매개변수이고, $A^{\pi_{\theta_t}}$ 는 시간 스텝 t 에서 이점 함수의 추정자이다. 그런 다음, PPO는 확률적 경사하강법을 통해 수학적 식 10의 PPO-clip 목표를 최대화하여 에이전트를 훈련하고 정책을 업데이트한다.

[0092] 두 제약 조건(C1 및 C2)을 위반하는 작업 선택은 불가능하다. 이를 고려하여 에이전트가 특정 동작을 선택하지 않도록 하는 동작 마스킹 기술을 사용하여 수렴에 도달하는 데 걸리는 시간을 줄인다. 따라서, j 번째 단계 시작 시 (S_j) 의 관찰 공간을 업데이트하기 전에 일부 행동을 마스킹하여 실행 가능한 행동 공간, A_j^{feasible} , ($A_j^{\text{feasible}} \subset A$)을 구축한다. j 번째 단계에 대해 각 작업 a_j ($a_j \in A$)가 선택되었다고 가정하면 두 가지 제약 조건, 즉 대역폭 제약 조건 $\max_{k=1, \dots, N^{\text{period}}} U_{a_j, k} \leq 1$ 과 저장 공간 제약 조건 ($U_{a_j}^{\text{st}} \leq 1$)를 검토한다. 이러한 두 가지 활용 제약 중 하나를 위반하는 모든 작업 a_j 는 작업 공간 A 에서 제거된다. 그런 다음, 에이전트는 알림을 받고 A_j^{feasible} 의 실행 가능 j 의 실행 가능한 작업 공간에서만 작업을 선택할 수 있다. 실시예에서는 포괄적인 DRL 알고리즘과 도구를 제공하고 다른 라이브러리보다 더 안정적인 것으로 알려진 Pytorch를 기반으로 하는 DRL 알고리즘의 오픈 소스 stable-baselines3 라이브러리를 사용하여 FPA를 구현할 수 있다. 이때, 안정적인 기준선에서 4개의 레이어로 완전히 연결된 신경망인 MlpPolicy 모델을 사용한다. 액션 마스킹을 구현하기 위해 stable-baselines3 contrib 저장소의 "InvalidActionEnvDiscrete" 함수를 사용한다. 도 5에 도시된 바와 같이, PPO의 기본 하이퍼 매개변수가 사용될 수 있다.

[0093] 도 6 및 도 7을 참고하면, 훈련 단계와 결정 단계로 구성된 FPA의 의사코드를 설명하기 위한 예이다. 훈련 단

계에서 PPO는 학습 효율성을 높이기 위해 지정된 에포크 수 동안 수집된 모든 궤적에 대해 미니배치 크기의 확률적 경사하강법을 사용하여 정책 π_{θ_t} 를 독립적으로 업데이트한다.

[0094] 실시예에서 제안된 비디오 파일 배치 방법을 평가하기 위해 시뮬레이션이 수행될 수 있다. 도 8에 도시된 표와 같이, 10개의 상용 SSD(Solid State Disk)의 전력, 대역폭 및 저장 용량 특성이 모델링될 수 있다. 각 DG는 4개의 SSD로 구성되어 비디오 스토리지 시스템에는 총 40개의 SSD가 있다. 실시예의 모델은 YouTube에서 권장하는 7가지 비트 전송률(1Mbps, 2.5Mbps, 5Mbps, 8Mbps, 16Mbps, 40Mbps 및 80Mbps)을 사용한다. 비디오 길이는 30분에서 120분 사이에서 무작위로 선택될 수 있다. 각 비디오의 인기도는 왜도(skewness) 매개변수가 0.3에서 0.7 사이인 Zipf 분포로 모델링될 수 있다. 여기서, 왜도 매개변수가 높을수록 특정 동영상에 대한 인기가 더 치우치는 경향이 있다. 총 요청 수는 일별 액세스 패턴을 따르며, 이는 시간에 따라 달라질 수 있다. 이를 허

용하기 위해 모든 DG(예를 들면, $\frac{\sum_{j=1}^{N_{file}} B_j^{file}}{\sum_{i=1}^{N_{DG}} B_i^{DG}}$)에 대한 총 대역폭 사용률은 평균이 0.6이고 표준 편차가 0.4인 정규 분포를 사용하여 생성되며 최대 사용률은 다음과 같이 제한될 수 있다. 0.8. FPA는 하루에 한 번 수행되어 해당 날짜의 작업량을 예측한다. 따라서 N_{period} 은 24로 설정되므로 총 대역폭 사용량은 하루 중 매 시간마다 변경될 수 있다. N^{file} 은 1000에서 1300 사이에서 무작위로 선택될 수 있다. 벤치마크로서 가장 인기 있는 것부터 시작하여 파일이 저장되는 DG 위치를 순서대로 결정하기 위해 다음 방법을 고려한다.

[0095] 1) RA(Random Allocation): 각 파일은 디스크 대역폭과 용량 조건이 만족될 때까지 무작위로 결정된 DG에 저장된다. 이는 Hadoop과 같은 분산 파일 시스템에서 일반적으로 사용되는 방법이다.

[0096] 2) Worst-Fit Allocation(WF): 파일은 남은 용량과 대역폭 사용률의 합이 가장 작은 DG에 순서대로 저장된다. 이는 bin-packing 문제에 자주 사용된다.

[0097] 3) Balanced Allocation(BA): 디스크 대역폭과 용량 활용도의 균형을 맞추기 위해 남은 용량과 대역폭 활용도의 차이가 가장 작은 DG가 다음 파일에 대해 먼저 선택된다.

[0098] 4) LPF(Lowest-Power-First Allocation): 각 DG에 각 파일을 저장할 때의 전력 소모량을 계산하고, 평균 전력 소모가 가장 낮은 DG를 먼저 할당한다.

[0099] 5) LPIF(Lowest-Power-Increase-First Allocation): LPF와 유사하지만 다음 파일에 대해 전력 소비 증가가 가장 작은 DG를 선택한다.

[0100] 6) LWPF(Lowest-Worst-Power-First Allocation): 각 DG의 각 기간에서 피크 전력을 최소화하는 것을 목표로 피크 기간 동안 가장 높은 전력을 소비하는 DG가 다음 파일로 선택된다.

[0101] 7) LWPIF(Lowest-Worst-Power-Increase-First Allocation): LWPF와 유사하지만 다음 파일에 대한 전력 소비 증가가 가장 낮은 DG를 선택한다.

[0102] 처음 3가지 방식(RA, WF, BA)은 대역폭과 스토리지 활용도를 높이기 위해 널리 사용되며, 나머지 4가지 방식(LPFI, LPIFI, LWPF, LWPIFI)에서는 전력 소모를 줄이는 것을 목표로 X_j 를 선택한다. 그런 다음 FPA를 다양한 시나리오에 대한 7가지 벤치마크 기법과 비교한다.

[0103] 제안한 기법은 타 기법보다 100번 비교 실험한 결과는 다음과 같다.

[0104] 미러링만 적용할 경우, 도 9를 참고하면, 평균 전력 감소 감소율을 보여주며, 다른 기법 대비 7.25%에서 16.2% (평균 10.36%) 전력 소모 감소율을 보여준다. 도 10을 참고하면, 최대 전력 감소율을 보여주며, 다른 기법 대비 12.61%에서 25.39% (평균 17.49)의 최대 전력 감소율을 보여준다.

[0105] 미러링과 트리플 중복을 동시에 적용할 경우(트리플 중복을 3개의 DG에 적용하고, 나머지 7개의 DG에는 미러링을 적용함), 도 11을 참고하면, 평균 전력 감소 감소율을 보여주며, 다른 기법 대비 9.83%에서 18.66% (평균 13.04%)의 전력 소모 감소율을 보여준다. 도 12를 참고하면, 최대 전력 감소율을 보여주며, 다른 기법 대비 14.75%에서 27.38%(평균 20.16%)의 최대 전력 감소율을 보여준다. 이러한 실험 결과들을 통해 실시예에서 제안한 기법이 언제나 우수함을 확인할 수 있다.

[0106] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로

세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0107] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

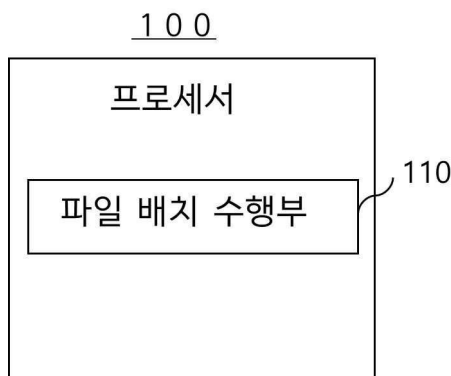
[0108] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic 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.

[0109] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

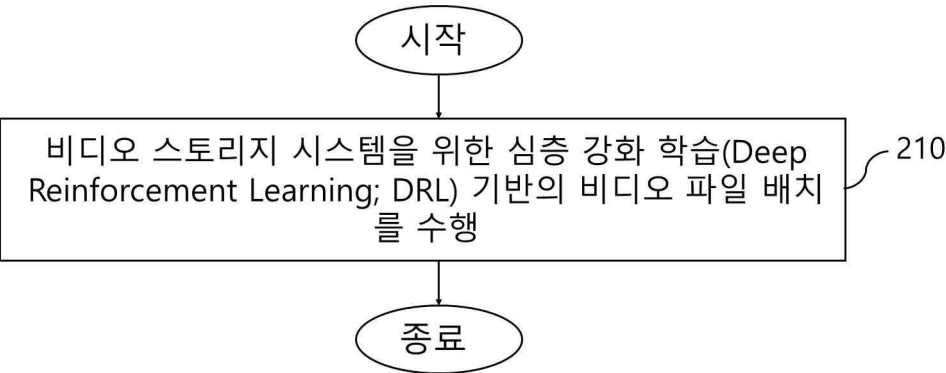
[0110] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

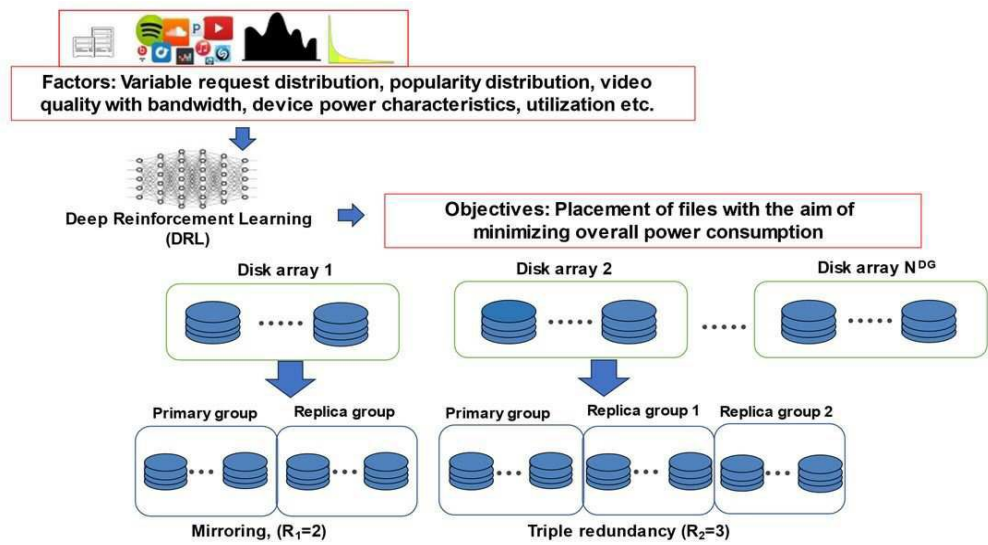
도면1



도면2



도면3



도면4

Primary group Replica groupPrimary group Replica groupPrimary group Replica group						
Disk index	1	2	3	4	5	6
Segments (S1-S6)	S1	S2	S3	S4	S5	S6
Segments (S1-S6)	S6	S1	S2	S3	S4	S5

도면5

TABLE I
PPO HYPER-PARAMETERS.

Hyper-parameter	Description	Values
Learning rate	Learning rate	0.0003
N_steps	Number of steps to be run per policy update	2048
Batch_size	Minibatch size	64
N_epochs	Number of epochs used to optimize loss	10
Clip_range	Clipping parameter	0.2
Gamma	Discount factor	0.99

도면6

Algorithm 1: FPA: training phase.

Input: $C_i^{\text{DG}}, B_i^{\text{DG}}, P_i^I, P_i^A, (i = 1, \dots, N^{\text{DG}})$
Output: PPO training model: $\mathcal{M}$

- 1 Constant values: $N^{\text{episode}};$
- 2 Initialize action space ($\{1, \dots, N^{\text{DG}}\}$) and observation space;
- 3 **for** $m = 1$ **to** $m = N^{\text{episode}}$ **do**
- 4 Build an observation space of $S_0;$
- 5 **for** $j = 1$ **to** $j = N^{\text{file}}$ **do**
- 6 Generate file-related workloads, $C_j^{\text{file}}, B_{j,k}^{\text{file}},$
 ($j = 1, \dots, N^{\text{file}}), (k = 1, \dots, N^{\text{period}});$
- 7 Calculate the feasible action space, $A_j^{\text{feasible}};$
- 8 Based on current policy by PPO (say, θ_t), action a_j
 is selected from $A_j^{\text{feasible}};$
- 9 Calculate the reward using Equation (8) for action
 $a_j;$
- 10 S_{j+1} is calculated based on utilizations and power
 as described in subsection V-B;
- 11 **end**
- 12 **end**
- 13 **return** PPO training model $\mathcal{M};$

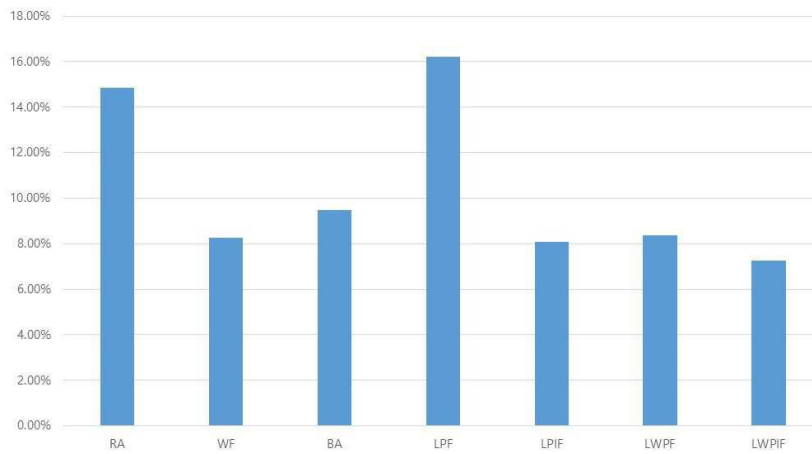
도면7

Algorithm 2: FPA: decision phase.	
Input: $\mathcal{M}$, C_i^{DG} , B_i^{DG} , P_i^I , P_i^A , ($i = 1, \dots, N^{\text{DG}}$), C_j^{file} , $B_{j,k}^{\text{file}}$, ($j = 1, \dots, N^{\text{file}}$), ($k = 1, \dots, N^{\text{period}}$)	
Output: $\forall j$, ($j = 1, \dots, N^{\text{file}}$), X_j	
1	Load PPO training model, $\mathcal{M}$;
2	Initialize file and DG workloads;
3	Build an observation space of S_0 ;
4	for $j = 1$ to $j = N^{\text{file}}$ do
5	Determine X_j for file j by model $\mathcal{M}$;
6	Update the observation space as a result of selection of X_j ;
7	Derive the feasible action space, A_j^{feasible} as a result of selection of X_j ;
8	end
9	return X_j , ($j = 1, \dots, N^{\text{file}}$);

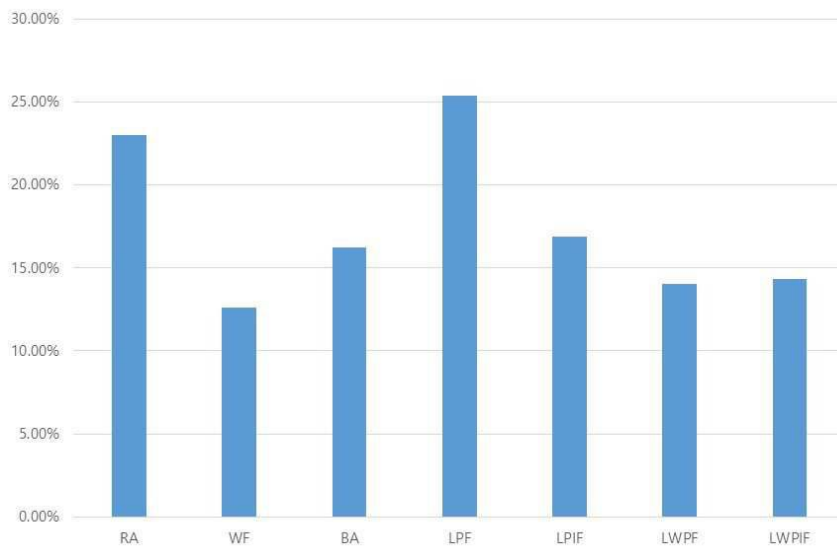
도면8

Vendor	Name	Bandwidth	Storage space	Active power	Idle power
Western Digital	Ultrastar DC SN655	15.3 GB/s	15.3TB	32	80
Western Digital	Ultrastar DC SN840	13.8 GB/s	15.3TB	24	72
Western Digital	WD Gold	7.3 GB/s	7.6TB	19.8	40
Seagate	Nytro 5350	14.0 GB/s	7.6TB	24	50
Micron Technology	Micron 9200	13.1 GB/s	12.8TB	32	100
Samsung	Samsung PM1725	13.1 GB/s	6.4TB	28	100
Samsung	Samsung PM1733	12.5 GB/s	7.6TB	34	80
Western Digital	Western Digital ultrastar	6.8 GB/s	6.4TB	12.8	50
Seagate	Nytro 5550	17.9 GB/s	6.4TB	24	56
Micron Technology	Micron 9400	23.4 GB/s	6.0TB	20	64

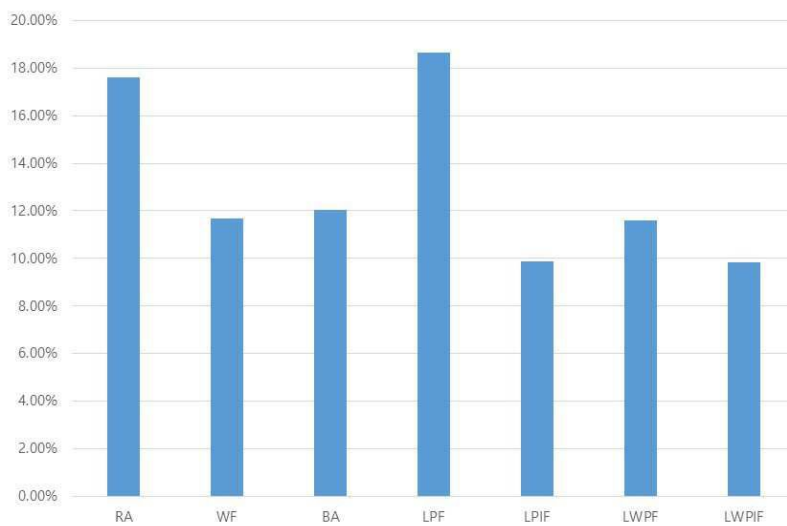
도면9



도면10



도면11



도면12

