



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년08월13일
(11) 등록번호 10-2289670
(24) 등록일자 2021년08월09일

(51) 국제특허분류(Int. Cl.)
H04N 21/262 (2011.01) H04N 21/2343 (2011.01)
H04N 21/2385 (2011.01)
(52) CPC특허분류
H04N 21/26233 (2013.01)
H04N 21/234381 (2013.01)
(21) 출원번호 10-2020-0041918
(22) 출원일자 2020년04월07일
심사청구일자 2020년04월07일
(56) 선행기술조사문헌
KR1020140058091 A*
*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)
이다영
인천광역시 미추홀구 인하로119번길 7, 406호(용현동, 청호주택)
(74) 대리인
양성보

전체 청구항 수 : 총 5 항

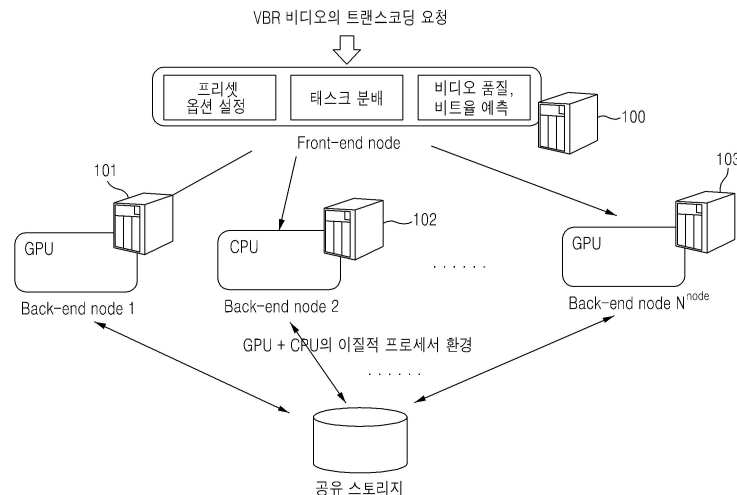
심사관 : 김건우

(54) 발명의 명칭 이기종 프로세서를 사용한 트랜스코딩 서버의 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링 기법

(57) 요약

이기종 프로세서를 사용한 트랜스코딩 서버의 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링 기법이 개시된다. 일 실시예에 따른 이기종 프로세서로 구성된 트랜스코딩 서버에서 수행되는 방법은, 가변 비트 전송률(VBR)에 기초하여 인코딩된 영상에 대한 비디오 트랜스코딩이 요청됨을 수신하는 단계; 상기 수신된 비디오 트랜스코딩의 요청에 따라 트랜스코딩 태스크를 배분하기 위한 트랜스코딩 정보를 예측하는 단계; 및 상기 예측된 트랜스코딩 정보의 결과를 이용하여 트랜스코딩 태스크가 할당될 노드 정보 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정하는 단계를 포함할 수 있다.

대표도



(52) CPC특허분류

H04N 21/2385 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 1711073299

과제번호 NRF-2017M3C4A7080248

부처명 과학기술정보통신부

과제관리(전문)기관명 한국연구재단

연구사업명 원천기술개발사업 (차세대정보컴퓨팅기술개발사업)

연구과제명 클라우드 환경에서 콘텐츠 및 사용자 속성을 고려한 저전력 멀티미디어 시스템 SW 개발

기 여 율 1/1

과제수행기관명 인하대학교

연구기간 2018.03.01 ~ 2020.12.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

이기종 프로세서로 구성된 트랜스코딩 서버에서 수행되는 방법에 있어서,

가변 비트 전송률(VBR)에 기초하여 인코딩된 영상에 대한 비디오 트랜스코딩이 요청됨을 수신하는 단계;

상기 수신된 비디오 트랜스 코딩의 요청에 따라 다중 회귀 분석을 통해 트랜스코딩 태스크를 배분하기 위한 트랜스코딩 정보를 예측하는 단계; 및

상기 예측된 트랜스 코딩 정보의 결과를 이용하여 트랜스코딩 태스크가 할당될 노드 정보 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정하는 단계

를 포함하고,

상기 결정하는 단계는,

상기 예측된 트랜스 코딩 정보의 결과에 포함된 각 프리셋 옵션 및 각 프로세서 노드에서 해상도의 비트율, 비디오 품질 및 트랜스코딩 시간을 이용하여 트랜스코딩 태스크가 할당될 GPU 또는 CPU 중 어느 하나의 프로세서 유형을 포함하는 프로세서 노드 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정하고, 최소 트랜스코딩 시간을 나타내는 노드와 프리셋 옵션을 선택하고, 트랜스코딩 증가 시간을 최소화하면서, 비트율 당 비디오 품질을 최대화하기 위하여 결정된 프로세서 노드에 워크로드를 재분배하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 트랜스 코딩 정보를 예측하는 단계는,

상기 영상의 각 세그먼트를 영상의 원본 데이터에 대한 상기 가변 비트 전송률을 사용하여 각 프리셋 옵션 및 각 프로세서 노드에서 각각의 해상도의 비트율 및 비디오 품질을 예측하고, 각각의 해상도의 트랜스코딩 시간을 순서대로 예측하는 단계

를 포함하는 방법.

청구항 3

제2항에 있어서,

상기 트랜스 코딩 정보를 예측하는 단계는,

다중 회귀 분석을 적용하여, 상기 영상의 원본 데이터의 비트 전송률을 통해 해상도와 비트율 및 비디오 품질의 관계, 해상도와 트랜스코딩 시간의 관계를 도출하는 단계

를 포함하는 방법.

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

제1항에 있어서,

상기 결정하는 단계는,

i (i 는 자연수)번째 태스크가 j (j 는 자연수)노드에서 특정 프리셋으로 실행될 때에 대한 유틸리티 값을 획득하고, 상기 획득된 유틸리티 값으로부터 기 설정된 기준 이상의 유틸리티 값을 이용하여 워크로드 재분배를 시행하는 단계

를 포함하는 방법.

청구항 8

이기종 프로세서로 구성된 트랜스코딩 서버에 있어서,

가변 비트 전송률(VBR)에 기초하여 인코딩된 영상에 대한 비디오 트랜스코딩이 요청됨을 수신하는 트랜스코딩 요청 수신부;

상기 수신된 비디오 트랜스 코딩의 요청에 따라 다중 회귀 분석을 통해 트랜스코딩 태스크를 배분하기 위한 트랜스코딩 정보를 예측하는 트랜스코딩 정보 예측부; 및

상기 예측된 트랜스 코딩 정보의 결과를 이용하여 트랜스코딩 태스크가 할당될 노드 정보 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정하는 결정부

를 포함하고,

상기 결정부는,

상기 예측된 트랜스 코딩 정보의 결과에 포함된 각 프리셋 옵션 및 각 프로세서 노드에서 해상도의 비트율, 비디오 품질 및 트랜스코딩 시간을 이용하여 트랜스코딩 태스크가 할당될 GPU 또는 CPU 중 어느 하나의 프로세서 유형을 포함하는 프로세서 노드 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정하고, 최소 트랜스코딩 시간을 나타내는 노드와 프리셋 옵션을 선택하고, 트랜스코딩 증가 시간을 최소화하면서, 비트율 당 비디오 품질을 최대화하기 위하여 결정된 프로세서 노드에 워크로드를 재분배하는

트랜스코딩 서버.

발명의 설명**기술 분야**

[0001] 아래의 설명은 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링 기술에 관한 것이다.

배경 기술

[0003] Dynamically adaptive streaming over HTTP(DASH)를 통한 동적 적응 스트리밍은 이제 Hulu, YouTube 및 Netflix를 비롯한 많은 주요 스트리밍 서비스 제공 업체가 채택함으로써 사실상 인터넷 비디오 스트리밍의 표준이 되었다. DASH는 각 비디오를 여러 비트율 버전으로 더 인코딩된 세그먼트로 나누며, 클라이언트 측 미디어 플레이어는 재생 중단을 최소화하기 위해 현재 네트워크 상태와 일치하는 적절한 비트 전송률 버전을 선택할 수 있다.

[0004] 트랜스코딩은 DASH에서 다양한 비트 전송률 버전을 생성하는 데 필수적이다. 예를 들어, Netflix 스트리밍에는 하나의 비디오 클립에 대해 120개의 트랜스코딩 작업(work)이 필요할 수 있다고 보고된 바 있다. 그러나 트랜스코딩은 계산 집약적이므로 서버에 상당한 처리 인프라가 필요하므로 서버 확장성에 부정적인 영향을 미치며, 증가하는 트랜스코딩에 대한 수요를 효과적으로 지원하기 위해 계산 단위를 점진적으로 추가해야 하므로 서버는 이기종 CPU와 GPU로 구성된다. 각 처리 장치는 비디오 품질, 트랜스코딩 속도 및 비용 측면에서 서로 다른 특성을 가지며, 트랜스코딩 작업 할당에 반영되어야 한다.

[0005] 인터넷 비디오 스트리밍은 자연스럽게 인터넷 비디오 트래픽의 양을 증가 시키며, 큰 규모의 서버 구조를 요구한다. 예를 들어, YouTube는 매일 4 천만 개의 동영상상을 제공하는 것으로 추정되는데, 이는 200TB의 데이터에

해당한다. 따라서 큰 규모의 트랜스코딩 서버가 필요하며 이에 대한 효과적인 대처가 중요하다.

- [0006] 최근 넷플릭스를 비롯한 스트리밍 회사들은 VBR(variable-bit-rate) 기술을 채택하기 시작했다. 이는 CBR(constant-bit-rate) 기술에 비해 네트워크 대역폭을 줄이면서 보다 나은 QoE(Quality of Experience)를 제공한다. 그러나, 가변 비트 전송률(VBR)을 활용한 DASH 기술의 대한 연구는 미미한 게 현실이다.
- [0007] 일반적으로 비디오 코덱에서는 트랜스코딩 시간, 비디오 품질 및 압축 비율 간의 트레이드 오프(trade-off)를 해결하기 위한 다양한 프리셋 옵션이 제공된다. 예를 들면, H.264 코덱은 9개의 프리셋 옵션 (ultrafast, superfast, veryfast, faster, fast, medium, slow, slower and veryslow)을 CPU에서 제공한다. 더 나은 비트율 당 비디오 품질을 제공하려면 고품질 사전 설정 옵션을 선택해야 하지만, 트랜스코딩 시간이 더 길어진다. 예를 들면, veryslow 프리셋 옵션은 최고의 비트율 당 비디오 품질을 제공하나, 트랜스코딩 시간이 길다는 문제가 있다. 따라서, 트레이드오프를 해결할 수 있는 프리셋 옵션의 선택이 중요하다.
- [0008] 업로드 후 빠른 스트리밍을 제공하려면 일부 작업을 가능한 빨리 트랜스코딩 해야 한다. 일부 영상의 트랜스코딩은 인기가 없기 때문에 지연될 수 있지만 일부 작업은 SLA (서비스 수준 계약)를 높이기 위해 가능한 한 빨리 실행해야 하며, 각 작업의 데드라인을 보장하기 위해 적절한 스케줄링을 해야 한다.
- [0009] 정리하자면, VBR 기반 트랜스코딩 서버에서 각 프리셋 및 프로세서 유형에 따른 트랜스코딩 시의 트랜스코딩 시간 및 트랜스코딩 결과물의 비디오 품질 및 비트율 예측 기법의 개발과, 트랜스코딩 작업의 인기 및 데드라인에 따른 프로세서 및 프리셋 할당 및 스케줄링 기법이 필요하다.
- [0010] 지금까지 DASH 시스템에서의 트랜스코딩 태스크 관리에 대한 연구가 지속적으로 진행되었으나, 이기종 프로세서로 구성된 트랜스코딩 서버에서 트랜스코딩 프리셋 옵션을 선택하고 데드라인 내에 VBR로 비디오를 트랜스코딩 하는 기술은 없었으며, VBR 기반 스트리밍 서버의 사용이 시작된 상황과, GPU와 CPU를 혼용해서 사용하기 시작한 이 시점에, 해당 기술의 개발은 필수적이다.

발명의 내용

해결하려는 과제

- [0012] 가변 비트 전송률(VBR, variable-bit-rate) 기술로 인코딩된 영상을 트랜스코딩하는 서버에서 트랜스코딩 데드라인을 만족하면서, 네트워크 대역폭을 줄이면서 더 나은 비디오 품질을 제공하기 위해서, 비트율 당 비디오 품질을 최대화하는 방법 및 장치를 제공할 수 있다. 이를 위해서 다중 회귀 분석을 통해서, 트랜스코딩 시간 및 비트율 당 비디오 품질을 예측하는 새로운 기술을 개발하고, 예측한 결과를 바탕으로, 태스크가 할당될 프로세서 노드 및 트랜스코딩을 위한 프리셋 옵션을 결정하는 새로운 기술을 제공할 수 있다.

과제의 해결 수단

- [0014] 이기종 프로세서로 구성된 트랜스코딩 서버에서 수행되는 방법은, 가변 비트 전송률(VBR)에 기초하여 인코딩된 영상에 대한 비디오 트랜스코딩이 요청됨을 수신하는 단계; 상기 수신된 비디오 트랜스 코딩의 요청에 따라 트랜스코딩 태스크를 배분하기 위한 트랜스코딩 정보를 예측하는 단계; 및 상기 예측된 트랜스 코딩 정보의 결과를 이용하여 트랜스코딩 태스크가 할당될 노드 정보 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정하는 단계를 포함할 수 있다.
- [0015] 상기 트랜스 코딩 정보를 예측하는 단계는, 상기 영상의 각 세그먼트를 영상의 원본 데이터에 대한 상기 가변 비트 전송률을 사용하여 각 프리셋 옵션 및 각 프로세서 노드에서 각각의 해상도의 비트율 및 비디오 품질을 예측하고, 각각의 해상도의 트랜스코딩 시간을 순서대로 예측하는 단계를 포함할 수 있다.
- [0016] 상기 트랜스 코딩 정보를 예측하는 단계는, 다중 회귀 분석을 적용하여, 상기 영상의 원본 데이터의 비트 전송률을 통해 해상도와 비트율 및 비디오 품질의 관계, 해상도와 트랜스코딩 시간의 관계를 도출하는 단계를 포함할 수 있다.
- [0017] 상기 결정하는 단계는, 상기 예측된 트랜스 코딩 정보의 결과에 포함된 각 프리셋 옵션 및 각 프로세서 노드에서 해상도의 비트율, 비디오 품질 및 트랜스코딩 시간을 이용하여 트랜스코딩 태스크가 할당될 GPU 또는 CPU 중 어느 하나의 프로세서 유형을 포함하는 프로세서 노드 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정하는 단계를 포함할 수 있다.
- [0018] 상기 결정하는 단계는, 최소 트랜스코딩 시간을 나타내는 노드와 프리셋 옵션을 선택하는 단계를 포함할 수 있

다.

[0019] 상기 결정하는 단계는, 트랜스코딩 증가 시간을 최소화하면서, 비트율 당 비디오 품질을 최대화하기 위하여 결정된 프로세서 노드에 워크로드를 재분배하는 단계를 포함할 수 있다.

[0020] 상기 결정하는 단계는, i (i 는 자연수)번째 태스크가 j (j 는 자연수)노드에서 특정 프리셋으로 실행될 때에 대한 유틸리티 값을 획득하고, 상기 획득된 유틸리티 값으로부터 기 설정된 기준 이상의 유틸리티 값을 이용하여 워크로드 재분배를 시행하는 단계를 포함할 수 있다.

[0021] 이기종 프로세서로 구성된 트랜스코딩 서버는, 가변 비트 전송률(VBR)에 기초하여 인코딩된 영상에 대한 비디오 트랜스코딩이 요청됨을 수신하는 트랜스코딩 요청 수신부; 상기 수신된 비디오 트랜스코딩의 요청에 따라 트랜스코딩 태스크를 배분하기 위한 트랜스코딩 정보를 예측하는 트랜스코딩 정보 예측부; 및 상기 예측된 트랜스코딩 정보의 결과를 이용하여 트랜스코딩 태스크가 할당될 노드 정보 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정하는 결정부를 포함할 수 있다.

발명의 효과

[0023] 이기종의 프로세서로 구성된 트랜스코딩 서버에서 트랜스코딩 프리셋 옵션을 선택하고 데드라인 내에 가변 비트 전송률에 기초하여 비디오를 트랜스코딩할 수 있다. 이에, 트랜스코딩 태스크의 데드라인을 만족시키면서 비트율 당 비디오 품질을 최대화할 수 있다.

도면의 간단한 설명

[0025] 도 1은 일 실시예에 따른 트랜스코딩 서버의 개괄적인 동작을 설명하기 위한 도면이다.

도 2는 일 실시예에 따른 트랜스코딩 서버에서 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링을 위한 알고리즘의 의사코드를 나타낸 것이다.

도 3은 일 실시예에 따른 트랜스코딩 서버의 구성을 설명하기 위한 블록도이다.

도 4는 일 실시예에 따른 트랜스코딩 서버에서 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링을 수행하는 방법을 설명하기 위한 흐름도이다.

도 5는 일 실시예에 따른 트랜스코딩 서버에서 제안된 방법과 타 태스크 할당 기법에서 비디오 품질 당 비트율을 트랜스코딩 요청의 도착률에 기초하여 비교한 결과를 설명하기 위한 그래프이다

도 6은 일 실시예에 따른 트랜스코딩 서버에서 제안된 방법과 타 태스크 할당 기법에서 비디오 품질 당 비트율을 데드라인 범위에 기초하여 비교한 결과를 설명하기 위한 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0026] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0028] 실시예에서는 이기종 CPU와 GPU로 구성된 비디오 트랜스코딩 서버(video transcoding server)에서의 각 트랜스코딩 태스크의 수행 데드라인을 만족시키면서, 비트율 당 비디오 품질(video quality per bitrate)을 최대화하는 새로운 기법을 설명하기로 한다.

[0029] 비트율 당 비디오 품질을 최대화하기 위해서는 프리셋(preset) 옵션과 프로세서의 유형에 따른 비디오 품질 값과, 트랜스코딩 시간을 정확하게 예측해야 한다. 이에, 실시예에서는 가변 비트 전송률(VBR, variable-bit-rate) 기술로 인코딩된 실제 비디오 클립을 사용하여 다중 회귀 기법(Multiple regression)을 통해 프리셋(preset) 매개 변수 및 프로세서의 유형에 따라 트랜스코딩 시간 및 비디오 품질과 비트율의 예상값을 정확하게 도출하는 방법 및 장치를 설명하기로 한다.

[0030] 상세하게는, 예측한 비디오 품질 값과 트랜스코딩 시간을 바탕으로, 비트율 당 비디오 품질을 최대화하는 새로운 기법을 제안하기로 한다. 실시예에서 제안하는 기법은 각 프로세서 노드 및 프리셋 옵션 선택에 대한 유틸리티(utility) 값을 정의하고, 유틸리티 값을 기준으로 비트 전송률 당 전체 비디오 품질을 최대화하기 위해 워크로드(작업부하)를 점차적으로 재분배할 수 있다. 아래에서, 회귀식은 정확하게 예측되며, 모든 트랜스코딩 태스크의 데드라인을 만족시키면서, 비트율 당 비디오 품질을 최대화함을 확인할 수 있다.

[0031] 도 1은 일 실시예에 따른 트랜스코딩 서버의 개괄적인 동작을 설명하기 위한 도면이다.

- [0032] 트랜스코딩 서버는 각각 다른 유형의 N^{node} 개의 노드로 구성되며 각 노드는 CPU 또는 GPU 노드이다. 각 노드에는 트랜스코딩 태스크가 할당될 수 있다. 전체 태스크의 숫자를 N^{task} 라고 한다. τ_i 를 i 번째 태스크라고 한다. τ_i 의 데드라인 초기값은 D_i^A 이다. 태스크의 데드라인은 상대적이므로 시간이 흐를수록 줄어든다. 알고리즘이 실행되는 현재 시간 cur 에서의 τ_i 의 데드라인을 D_i^R 라고 하고, $D_i^R = D_i^A - cur$ 이다.
- [0033] 대쉬(DASH)에서 비디오는 세그먼트로 나뉘며 각 세그먼트는 서로 다른 비디오 품질의 여러 개의 해상도 버전으로 트랜스코딩 될 수 있다. 이때, 각 노드는 프로세서의 타입과, 프로세서에서 사용하는 코덱에 따라 프리셋 옵션수가 다르다. N_j^{para} ($j=1, \dots, N^{node}$)는 노드 j 의 프리셋 옵션 수이다. 대부분의 트랜스코딩을 위한 코덱은 다양한 트랜스코딩 속도, 비디오 품질 및 압축 비율을 생성하는 프리셋 옵션 모음을 제공할 수 있다.
- [0034] 태스크 τ_i 가 노드 j 에서 프리셋 k 로 실행되어 생성할 때 소요되는 시간을 $T_{i,j,k}$, 트랜스코딩된 버전의 비트율을 $B_{i,j,k}$, 비디오 품질 인덱스를 $Q_{i,j,k}$ 라고 한다. 실시예에서는 SSIM 지수를 사용하여 비디오 품질이 표현될 수 있다. 프리셋 옵션은 $Q_{i,j,k}$ 값의 오름차순으로 정렬되어 있다고 가정하기로 한다.
- [0035] 트랜스코딩이 완료되면 비디오 클립이 사용자에게 스트리밍될 수 있다. 비디오 스트리밍의 인기도는 3단계로 구분될 수 있다. 예를 들면, 처음 출시될 때의 핫(hot) 단계, 중간 정도의 웜(warm) 단계, 동영상에 노출되지 않을 때의 콜드(cold) 단계로 구분될 수 있다. 이러한 비디오의 인기는 머신러닝 기술이 사용되어 총 액세스 숫자가 예측될 수 있다. τ_i 의 액세스 숫자를 N_i^{access} 라고 정의한다.
- [0036] 비디오 인기도를 고려하여 태스크 τ_i 가 노드 j 에서 프리셋 k 로 실행되어 생성된 트랜스코딩된 버전의 비트율 당 비디오 품질을 $G_{i,j,k}$ 라고 하면, $G_{i,j,k} = \frac{Q_{i,j,k} N_i^{access}}{B_{i,j,k}}$ 이다.
- [0037] 각 노드는 D_i^R 가 가장 낮은 태스크 τ_i 를 가장 먼저 트랜스코딩하는 EDF 스케줄링을 사용할 수 있다. 그러므로 노드 j 에 할당된 태스크의 리스트인 A_j 는 D_i^R 의 값에 따라 오름차순으로 정렬될 수 있다. $L_{i,j}$ 는 태스크 τ_i 가 노드 j 에 할당되었을 때, 태스크 τ_i 보다 데드라인이 빠른 태스크들의 리스트를 의미한다. Jackson의 정리에 따라 노드 j 에 할당된 모든 태스크들의 데드라인을 만족시키기 위하여, $\forall \tau_i \in A_j, \sum_{\tau_m \in L_{i,j}} T_{m,X_m,Y_m} \leq D_m$ 을 만족해야 한다.
- [0038] 실시예에서는 전체적인 비트율 당 비디오 품질 $\sum_{i=1}^{N^{task}} G_{i,X_i,Y_i}$ 을 최대화하면서, 데드라인을 만족시키기 위해 각 태스크 τ_i ($i=1, \dots, N^{task}$)가 할당될 노드 X_i 및 프리셋 Y_i 를 선택하는 방법에 대하여 설명하기로 한다. 상기 전체적인 비트율 당 비디오 품질 $\sum_{i=1}^{N^{task}} G_{i,X_i,Y_i}$ 을 최대화하면서, 데드라인을 만족시키기 위해 각 태스크 τ_i ($i=1, \dots, N^{task}$)가 할당될 노드 X_i 및 프리셋 Y_i 를 선택하는 문제는 아래와 같이 표현될 수 있다.

$$\begin{aligned} & \text{Maximize } \sum_{i=1}^{N^{task}} G_{i, X_i, Y_i} \\ & \text{subject to } \forall \tau_i \in A_j, \sum_{\tau_m \in L_{i,j}} T_{m, X_m, Y_m} \leq D_m \\ & j = 1, \dots, N^{node} \end{aligned}$$

[0039]

[0040]

실시예에서는 상기 문제를 해결하기 위해 트랜스코딩 시간, 비디오 품질 및 비트율과 같은 알고리즘의 파라미터를 예측하는 식을 다중 회귀를 통하여 도출하고, 최적의 값에 근접하는 휴리스틱 기반의 알고리즘을 제안하기로 한다.

[0041]

트랜스코딩 서버는 다중 회귀를 통한 파라미터를 예측할 수 있다. 트랜스코딩 서버는 트랜스코딩 시간, 비디오 품질 및 비트율을 포함하는 파라미터를 예측할 수 있다. 각 세그먼트를 가변 비트 전송률(VBR) 기법을 사용하여 트랜스코딩될 수 있다. 각 프리셋 옵션에는 압축률이 다르므로 동일한 버전일지라도 세그먼트가 다르면 비트율도 다르게 된다. 이에 따라, 각 프로세서에 트랜스코딩 작업을 효율적으로 배분하기 위하여 비트율, 비디오 품질 및 트랜스코딩 시간과 같은 파라미터가 예측되어야 한다. 실시예에서는 트랜스코딩 서버는 비트율, 비디오 품질 및 트랜스코딩 시간을 예측하기 위하여 복수 개의 해상도 각각에 대한 복수 개의 기 설정된 길이의 비디오 세그먼트에 대한 비트율, SSIM 및 트랜스코딩 시간을 측정할 수 있다. 또한, 샘플의 프레임 속도, 픽셀 당 비트도 각각 기 설정된 값으로 설정될 수 있다. 예를 들면, 240p, 360p, 480p 및 720p의 3000개의 6초 길이의 비디오 세그먼트 (6초 길이)에 대한 비트율, SSIM값 및 트랜스코딩 시간이 측정될 수 있다. 샘플의 프레임 속도는 30fps로, 픽셀 당 비트(bpp)는 0.15로 설정될 수 있다. 또한, 다양한 버전의 CPU, GPU의 유형의 노드가 고려될 수 있다. 예를 들면, Intel i79700 CPU, NVIDIA GTX2070 GPU의 두 가지 유형의 노드를 고려할 수 있다. 이때, CPU는 H264 코덱이 사용되며, 코덱의 복수 가지(예를 들면, 9가지) 프리셋 옵션(ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow)을 사용하여 각 세그먼트가 트랜스코딩될 수 있다. GPU는 H264_CUVID 코덱을 통해 디코딩, H264_NVENC 코덱을 통해 인코딩을 진행하며, 코덱의 복수 가지(예를 들면, 4가지)의 프리셋(LLHP, HP, LLHQ, HQ)을 사용하여 각 세그먼트가 트랜스코딩될 수 있다. 프로세서 노드는 각 세그먼트 당 복수 개(예를 들면, 52개)의 트랜스코딩 작업을 실행하여 복수 개(예를 들면, 4개)의 트랜스코딩 된 버전을 생성할 수 있다. 회귀 분석에는 총 156,000개의 트랜스코딩 작업이 필요하며 이러한 트랜스코딩 작업이 스크립트로 자동화될 수 있다.

[0042]

다중 회귀를 위한 해상도 $V_l (l=1, \dots, N_r)$ 를 나타내는 더미 변수를 설명하기로 한다. 여기서, N_r 는 다음과 같이 트랜스코딩 된 총 해상도의 수이다.

$$V_l = \begin{cases} 1 & \text{Selected resolution index is } l \\ 0 & \text{Otherwise.} \end{cases}$$

[0043]

[0044]

원본 비디오의 해상도가 1080p이고, 4개의 트랜스코딩된 해상도(240p, 360p, 480p 및 720p)가 존재한다고 가정하기로 한다. N_r 는 4이고 $(V_1, V_2, V_3, V_4) = (0, 0, 0, 1)$ 은 720p를 나타내는 더미 변수이다. 트랜스코딩 전에는 요청한 비디오 세그먼트의 최고 해상도(또는 원본 해상도)에 대한 비트 전송률(kbps) H_1 의 값만 알려져 있다.

[0045]

트랜스코딩 서버는 최고 해상도(또는 원본 해상도)에 대한 비트 전송률 H_1 의 값만으로도, 각 프리셋 옵션 및 각 프로세서 노드에서의 모든 해상도의 비트율 및 비디오 품질을 예측하고, 모든 해상도의 트랜스코딩 시간을 순서대로 예측할 수 있다. 이를 위하여, 다중 회귀 분석을 적용하며, 오직 최고 해상도인 원본의 비트 전송률만으로 해상도와 비트율 및 비디오 품질 관계, 해상도와 트랜스코딩 시간의 관계를 도출할 수 있다.

[0046]

트랜스코딩 서버는 비트율과 비디오 품질(SSIM)을 예측할 수 있다. 트랜스코딩 전에 원본 버전 (1080p)의 비트 전송률과 원본 버전의 SSIM 값이 1이라는 정보를 사용할 수 있다. 트랜스코딩 서버는 해상도 l 의 비트율을 획득하기 위해 다중 회귀 기술을 사용하여 $l+1$ 의 비트율 및 SSIM 값을 사용하여 하위 트랜스코딩된 버전 해상도의 비트율과 비디오 품질을 도출할 수 있다. 노드 j 및 프리셋 k 에서 태스크 τ_i 에 의해 트랜스코딩 된 세그먼트

의 해상도를 I 라고 가정할 경우, $B_{i,j,k}^{up}$ 는 해당 비디오 세그먼트의 해상도 $I+1$ 의 비트 전송률을 나타내고, $Q_{i,j,k}^{up}$ 는 해당 비디오 세그먼트의 해상도 $I+1$ 의 SSIM 값을 나타낸다.

[0047] 노드 j 및 프리셋 k 에서 태스크 τ_i 에 의해 트랜스코딩된 결과의 비트율은 아래의 식을 따른다.
 $\alpha_{j,k}^1 \sim \alpha_{j,k}^{N_r+3}$ 값은 회귀식의 파라미터이다.

[0048]
$$B_{i,j,k} = \exp(\alpha_{j,k}^1 + \alpha_{j,k}^2 \ln B_{i,j,k}^{up} + \alpha_{j,k}^3 Q_{i,j,k}^{up} + \alpha_{j,k}^4 V_1 + \dots + \alpha_{j,k}^{N_r+3} V_{N_r})$$

[0049] 트랜스코딩 서버는 CPU 트랜스코딩 시간을 예측할 수 있다. 비트율과 비디오 품질(SSIM)을 통하여 획득된 각 해상도에서 비트 레이트 및 SSIM값을 사용하여 CPU 트랜스코딩 시간을 도출하기 위한 다중 회귀식을 설정할 수 있다. 이에, 설정된 다중 회귀식은 다음과 같이 나타낼 수 있다. $\gamma_{j,k}^1 \sim \gamma_{j,k}^{N_r+3}$ 값은 회귀식의 파라미터이다.

[0050]
$$T_{i,j,k} = \gamma_{j,k}^1 + \gamma_{j,k}^2 \ln H_i + \gamma_{j,k}^3 \ln B_{i,j,k} + \gamma_{j,k}^4 Q_{i,j,k} + \gamma_{j,k}^5 V_1 + \dots + \gamma_{j,k}^{N_r+4} V_{N_r}$$

[0051] 트랜스코딩 서버는 GPU 트랜스코딩 시간을 예측할 수 있다. 회귀 분석에 따르면, 비트율, SSIM, 해상도 및 GPU 트랜스코딩 시간 간의 관계가 약하다. 예를 들면, 모든 프리셋 옵션에 대해 결정 계수 R 는 0.586보다 작다. 이에 따라 파라미터(매개변수) 값으로 평균 GPU 트랜스코딩 시간을 사용할 수 있다.

[0052] 도 2는 일 실시예에 따른 트랜스코딩 서버에서 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링을 위한 알고리즘의 의사코드를 나타낸 것이다.

[0053] 트랜스코딩 서버는 트랜스코딩 태스크가 할당된 프로세서 노드 및 트랜스코딩을 위한 프리셋 옵션을 결정할 수 있다. 트랜스코딩 서버는 초기화 및 워크로드 재분배를 포함하는 두 단계를 수행할 수 있다. 초기화 단계에서, 최소 트랜스코딩 시간을 나타내는 노드와 프리셋 옵션을 선택하고, 재분배 단계에서, 트랜스코딩 증가 시간을 최소화하면서, 비트율 당 비디오 품질을 최대화하는 방향으로 워크로드를 재분배할 수 있다.

[0054] 초기화 단계에 대하여 상세하게 설명하기로 한다. X_i^{temp} 와 Y_i^{temp} 는 각각 X_i 와 Y_i 의 임시값을 저장하는 값이다. X_i^{temp} 및 Y_i^{temp} 의 모든 값은 가장 낮은 트랜스코딩이 이루어지는 노드 X_i^{low} 와 프리셋 Y_i^{low} 로 초기화할 수 있다. 예를 들면, CPU 노드에 프리셋을 ultrafast 옵션으로 모두 할당할 수 있다. 이는 모든 프로세서의(GPU 포함) 모든 프리셋과 비교했을 때, 가장 빠른 트랜스코딩 시간을 가지지만, 비디오 품질이 굉장히 낮아질 수 있다.

[0055] 워크로드 재분배 단계에서 대하여 상세하게 설명하기로 한다. X_i^{temp} 와 Y_i^{temp} 의 값은 $\sum_{i=1}^{N^{task}} G_{i,X_i,Y_i}$ 를 최대화하기 위한 값으로 변경될 수 있다. τ_i 태스크가 j 노드에서 프리셋 k 로 실행될 때에 대한 유틸리티 값 $U_{i,j,k}(i=1, \dots, N^{task})(j=1, \dots, N^{node})(k=1, \dots, N_j^{para})$ 를 정의할 수 있다. 이때, $U_{i,j,k}$ 의 (j, k) 가 (X_i^{low}, Y_i^{low}) 일 경우는 제외하기로 한다. $U_{i,j,k}$ 는 다음과 같이 나타낼 수 있다.

[0056]
$$U_{i,j,k} = \frac{G_{i,j,k} - G_{i,X_i^{low}, Y_i^{low}}}{T_{i,X_i^{low}, Y_i^{low}}^{slack} - T_{i,j,k}^{slack}}$$

[0057] $U_{i,j,k}$ 값에서 분자는 (j, k) 가 (X_i^{low}, Y_i^{low}) 와 비교할 때 대비 비트율 당 비디오 품질의 증가를 나타내며, 분

모는 데드라인 여유 시간 감소를 나타낸다. 다시 말해서, 트랜스코딩 서버는 트랜스코딩 증가 시간을 최소화하면서, 비트율 당 비디오 품질을 최대화하는 방향을 선택하기 위하여, $U_{i,j,k}$ 값이 높은 선택을 선호하여 워크로드 재분배를 시행할 수 있다.

- [0058] 워크로드 재분배 단계는 전체 비디오 품질을 높이는 것이 목표이다. 이를 위해 생성된 모든 유틸리티 값 $U_{i,j,k}$ 는 집합 A 에 저장되며, A 에서 가장 높은 값 $U_{I^{task}, I^{node}, I^{para}}$ 를 선택하고 아래의 조건이 만족되는지 확인할 수 있다.

$$G_{I^{task}, X_{I^{task}}^{temp}, Y_{I^{task}}^{temp}} < G_{I^{task}, I^{node}, I^{para}}$$

[0059]

상기 조건이 만족할 경우, Jackson의 정리에 따라 노드 j 에 할당된 태스크들의 데드라인이 충족시키기 위해

$\forall \tau_i \in A_j, \sum_{\tau_m \in L_{i,j}} T_{m, X_m, Y_m} \leq D_m$ 를 만족하는지 확인한 후 $X_{I^{node}}^{temp}$ 와 $Y_{I^{node}}^{temp}$ 를 I^{node} , I^{para} 로 업데이트할 수 있다. 이후 A 에서 $U_{I^{task}, I^{node}, I^{para}}$ 를 삭제하고 집합 A 가 빌 때까지 A 에서 가장 높은 값 $U_{I^{task}, I^{node}, I^{para}}$ 를 선택하는 일을 반복할 수 있다. 집합 A 가 모두 비게 되면 마지막으로, X_i 와 Y_i 의 값은 X_i^{temp} 와 Y_i^{temp} 으로 업데이트될 수 있다.

- [0061] 도 3은 일 실시예에 따른 트랜스코딩 서버의 구성을 설명하기 위한 블록도이고, 도 4는 일 실시예에 따른 트랜스코딩 서버에서 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링을 수행하는 방법을 설명하기 위한 흐름도이다.

- [0062] 트랜스코딩 서버(100)에 포함된 프로세서는 트랜스코딩 요청 수신부(310), 트랜스코딩 정보 예측부(320) 및 결정부(330)를 포함할 수 있다. 이러한 프로세서 및 프로세서의 구성요소들은 도 4의 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링을 수행하는 방법이 포함하는 단계들(410 내지 430)을 수행하도록 트랜스코딩 서버를 제어할 수 있다. 이때, 프로세서 및 프로세서의 구성요소들은 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다. 여기서, 프로세서의 구성요소들은 트랜스코딩 서버(100)에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다.

- [0063] 프로세서는 비디오 품질 최대화를 위한 태스크 할당 및 스케줄링을 수행하는 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩할 수 있다. 예를 들면, 트랜스코딩 서버(100)에서 프로그램이 실행되면, 프로세서는 운영체제의 제어에 따라 프로그램의 파일로부터 프로그램 코드를 메모리에 로딩하도록 트랜스코딩 서버를 제어할 수 있다.

- [0064] 단계(410)에서 트랜스코딩 요청 수신부(310)는 가변 비트 전송률(VBR)에 기초하여 인코딩된 영상에 대한 비디오 트랜스코딩이 요청됨을 수신할 수 있다.

- [0066] 단계(420)에서 트랜스코딩 정보 예측부(320)는 수신된 비디오 트랜스 코딩의 요청에 따라 트랜스코딩 태스크를 배분하기 위한 트랜스코딩 정보를 예측할 수 있다. 트랜스코딩 정보 예측부(320)는 영상의 각 세그먼트를 영상의 원본 데이터에 대한 가변 비트 전송률을 사용하여 각 프리셋 옵션 및 각 프로세서 노드에서 각각의 해상도의 비트율 및 비디오 품질을 예측하고, 각각의 해상도의 트랜스코딩 시간을 순서대로 예측할 수 있다. 트랜스코딩 정보 예측부(320)는 다중 회귀 분석을 적용하여, 영상의 원본 데이터의 비트 전송률을 통해 해상도와 비트율 및 비디오 품질의 관계, 해상도와 트랜스코딩 시간의 관계를 도출할 수 있다.

- [0067] 단계(430)에서 결정부(330)는 예측된 트랜스 코딩 정보의 결과를 이용하여 트랜스코딩 태스크가 할당될 노드 정보 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정할 수 있다. 결정부(330)는 예측된 트랜스 코딩 정보의 결과에 포함된 각 프리셋 옵션 및 각 프로세서 노드에서 해상도의 비트율, 비디오 품질 및 트랜스코딩 시간을 이용하여 트랜스코딩 태스크가 할당될 GPU 또는 CPU 중 어느 하나의 프로세서 유형을 포함하는 프로세서 노드 및 트랜스코딩을 위한 프리셋 옵션 정보를 결정할 수 있다. 결정부(330)는 최소 트랜스코딩 시간을 나타내는 노드와 프리셋 옵션을 선택할 수 있다. 결정부(330)는 트랜스코딩 증가 시간을 최소화하면서, 비트율 당 비디오 품질

을 최대화하기 위하여 결정된 프로세서 노드에 워크로드를 재분배할 수 있다. 결정부(330)는 i (i 는 자연수)번째 태스크가 j (j 는 자연수)노드에서 특정 프리셋으로 실행될 때에 대한 유틸리티 값을 획득하고, 획득된 유틸리티 값으로부터 기 설정된 기준 이상의 유틸리티 값을 이용하여 워크로드 재분배를 시행할 수 있다.

[0068] 도 5는 일 실시예에 따른 트랜스코딩 서버에서 제안된 방법과 타 태스크 할당 기법에서 비디오 품질 당 비트율을 트랜스코딩 요청의 도착률에 기초하여 비교한 결과를 설명하기 위한 그래프이고, 도 6은 일 실시예에 따른 트랜스코딩 서버에서 제안된 방법과 타 태스크 할당 기법에서 비디오 품질 당 비트율을 데드라인 범위에 기초하여 비교한 결과를 설명하기 위한 그래프이다.

[0069] 유튜브, 트위치 등 소셜 스트리밍 및 개인 브로드캐스팅 활성화로 인한 트랜스코딩 서버 수요의 폭증으로 현재 한계에 봉착하였다. 예를 들면, 트위치는 공격적으로 서버 확장에 나섬에도 불구하고, 많은 영상의 트랜스코딩 지원이 어려운게 현실이다. 또한 넷플릭스를 비롯한 많은 스트리밍 회사가 가변 비트 전송률(VBR) 인코딩 기법을 도입하고, GPU와 CPU를 혼용 사용하기 시작하였다. 이와 같은 배경임에도 불구하고, 실시예에서 제안된 이기종 프로세서로 구성된 트랜스코딩 서버는 트랜스코딩 프리셋 옵션을 선택하고 데드라인 내에 VBR로 비디오를 트랜스코딩을 수행할 수 있다.

[0070] 실시예에서 제안된 이기종 프로세서로 구성된 트랜스코딩 서버의 동작에 따른 효과를 평가하기 위하여 실제 비디오 세트를 사용하여 트랜스코딩 시간 및 비트율 당 비디오 품질의 회귀 분석 정확도를 분석한 다음, 초당 트랜스코딩 요청 횟수 및 데드라인의 범위설정이 미치는 영향을 확인하기 위해 데드라인 미스(miss) 횟수 및 스트리밍된 비디오 품질을 시뮬레이션을 통해서 측정될 수 있다.

[0071] 회귀 분석의 정확도가 도출될 수 있다. 도출한 회귀식의 정확도 확인을 위해 회귀식 도출에 사용되지 않은 복수 개(예를 들면, 1000 개)의 서로 다른 비디오 세그먼트가 특정 해상도(예를 들면, 1080p)에서 4 개의 다른 해상도 버전 (240p, 360p, 480p 및 720p)으로 트랜스코딩되고, SSIM 값, 트랜스 코딩 시간 및 비트율이 측정될 수 있다. 실제 값과 회귀로 획득된 예상 값 사이의 MAPE(mean absolute percentage error) 값이 계산될 수 있다. 이는 스케일 독립성과 해석성의 이점으로 인해 가장 널리 사용되는 예측 정확도 측정 방법 중 하나이다. MAPE의 값이 10보다 작으면 매우 정확한 예측, 20보다 작으면 좋은 예측으로 간주된다.

[0072] 표 1은 전체 MAPE를 요약한 것으로, 앞서 말했듯 프리셋 옵션은 프리셋을 통하여 획득될 수 있는 비디오 품질 값의 오름차순으로 정렬되어 있으며, 트랜스코딩 시간, 비트율, 비디오 품질이 매우 정확하게 예측됨을 확인할 수 있다.

[0073] 표 1: 각 프리셋 옵션에서의 MAPE 값

프리셋	1	2	3	4	5	6	7	8	9
CPU $B_{i,j,k}$	6.593	4.245	5.451	5.432	5.216	5.321	5.456	5.399	5.609
GPU $B_{i,j,k}$	10.878	10.615	11.072	10.236	-	-	-	-	-
CPU $Q_{i,j,k}$	2.016	1.52	1.399	1.366	1.349	1.343	1.336	1.316	1.312
GPU $Q_{i,j,k}$	1.658	1.628	1.627	1.607	-	-	-	-	-
CPU $T_{i,j,k}$	3.548	3.871	4.383	5.393	5.872	6.144	6.504	9.674	14.727

[0074] 시뮬레이션이 수행된 결과가 도출될 수 있다. 비트율 당 최종 비디오 품질 및 데드라인 미스 횟수에 대한 제안한 발명의 영향을 조사하기 위해 시뮬레이션이 수행될 수 있다. 비디오 품질은 트랜스코딩을 끝낸 이후 따로 초당 복수 개(예를 들면, 100)개의 버전을 기 설정된 시간(예를 들면, 24시간) 동안 스트리밍되어 도출될 수 있다.

[0076] 시뮬레이션 환경은 다음과 같이 설정될 수 있다.

[0077] 비디오 원본의 비트율 정보: 가변 비트 전송률(VBR)의 비트율은 정규회귀를 따르므로, 회귀에 사용한 3,000개의 1080p의 실제 비디오 세그먼트의 평균 비트율과 표준편차를 사용하여 1080p 비디오 클립의 비트율이 랜덤으로 생성될 수 있다. 각 클립의 길이는 1분에서 1시간 사이이며 각 세그먼트의 길이는 6초로 설정될 수 있다.

[0078] 인기도 설정: 비디오 인기는 일반적으로 Zipf 분포를 따른다. 각 비디오 클립의 인기도는 $\theta = 0.271$ 로 설정될 수 있다. 또한 비디오 시청이 중간에 중단될 수 있으므로 각 비디오의 세그먼트의 인기 역시 $\theta 0.2$ 인 zipf 분포를 따르며, 이는 앞의 세그먼트가 더 인기가 높다는 것을 의미한다.

- [0079] P_l 를 해상도 $l(l=1, \dots, N_r)$ 의 인기도라고 정의하기로 한다. 이때, 240p, 360p, 480p, 720p, 1080p의 인기도인 $(P_1, P_2, P_3, P_4, P_5)$ 의 조합 3가지를 고려했으며, 각각 HRP(high resolutions are popular), MRP(medium resolutions are popular), LRP(high resolutions are popular)라고 한다. 각 조합의 인기도는 아래와 같다.
- [0080] HRP: $(P_1, P_2, P_3, P_4, P_5) = (0.1, 0.1, 0.2, 0.3, 0.3)$
- [0081] MRP: $(P_1, P_2, P_3, P_4, P_5) = (0.1, 0.25, 0.3, 0.25, 0.1)$
- [0082] LRP: $(P_1, P_2, P_3, P_4, P_5) = (0.3, 0.3, 0.2, 0.1, 0.1)$.
- [0083] 위와 같은 설정을 바탕으로 각 태스크 τ_i 의 N_i^{access} 가 도출될 수 있다.
- [0084] 트랜스코딩 환경: 트랜스 코딩 요청 수는 초당 평균 요청이 λ^{trans} 로 설정되는 포아송 분포를 따르는 것으로 가정하며, 트랜스코딩 요청은 총 24시간 동안 진행될 수 있다. 각 태스크의 할당 및 프리셋 선택은 주기적으로 1초마다 실행될 수 있다. N^{node} 는 10이며 Intel i79700 CPU, NVIDIA GTX2070 GPU를 사용한다고 가정하기로 한다.
- [0085] 비교 대상 기법: 실시예에서 제안된 방법의 효과를 확인하기 위해 트랜스코딩 결과를 각각의 태스크 할당 방법 및 프리셋 옵션 선택 방법들의 조합과 비교될 수 있다. 이때, 태스크 할당 방법은 Round-robin(RR)과 Worst-fit(WF) 두 가지이다.
- [0086] 라운드 로빈(RR)은 각 태스크를 라운드 로빈 방식으로 노드에 할당하는 방식이다. 다시 말해서, 태스크 τ_i 는 $i \% N^{\text{node}}$ 에 할당될 수 있다($i \% N^{\text{node}} = 0$ 일 경우 N^{node} 에 할당된다.)
- [0087] 워스트 핏(WF)은 각 노드에 할당된 태스크들의 데드라인의 합의 균형을 맞추어 할당하는 방식이다. 다시 말해서, 새로운 태스크는 데드라인의 합이 가장 낮은 노드에 할당될 수 있다.
- [0088] 프리셋 옵션 선택 방법은 Highest quality First(HF), Lowest transcoding time First(LF), Default preset First(DF)를 포함할 수 있다.
- [0089] HF은 전체 비디오 품질을 최대화하는 것을 목표로 하므로, CPU에서는 veryslow, GPU에서는 HQ 옵션을 선택한다.
- [0090] LF은 전체 트랜스 코딩 시간을 최소화하기 위해 CPU에서는 ultrafast를 선택한다. GPU에서는 실제 트랜스코딩을 통해 측정한 결과, 프리셋에 따른 트랜스코딩 시간차이가 무의미함이 확인될 수 있다. 가장 비디오 품질이 높은 HQ 옵션을 선택한다.
- [0091] DF는 각 프로세서의 디폴트 옵션을 선택한다. 디폴트 프리셋 옵션은 CPU의 경우 medium, GPU의 경우 HQ이다.
- [0092] 다시 말해서, 실시예에서 제안된 방법과의 비교 기법은 두 가지 태스크 할당 방법(RR, WF)과 세 가지 프리셋 옵션 선택 방법(HF, LF, DF)을 결합하는 6가지 (RR+HF, WF+HF, RR+LF, WF+LF, RR+DF 및 WF+DF)이다.
- [0093] 데드라인 미스 회수와 비트율 당 비디오 품질이 비교될 수 있다. λ^{trans} (트랜스코딩 요청의 도착률)에 따른 결과에 대하여 설명하기로 한다. 트랜스코딩 요청의 도착률과 데드라인의 범위를 변화시키면서 다양한 환경에 대한 실험이 진행될 수 있다. 먼저 트랜스코딩 서버의 워크로드의 변화가 비트율 당 비디오 품질 및 데드라인 미스에 미치는 영향에 대하여 실험이 수행될 수 있다. 트랜스 코딩 서버의 로드는 N^{node} 및 λ^{trans} 에 의해 결정된다. N^{node} 가 증가하거나 λ^{trans} 가 감소하게 되면 트랜스 코딩 서버의 로드가 감소한다. 이때, λ^{trans} 의 숫자를 조절하고, N^{node} 를 10으로 고정하여 실험이 진행될 수 있다. 데드라인의 초기값은 3초에서 20 초 사이이다.
- [0094] 도 3는 λ^{trans} 값에 따른 비트율 당 비디오 품질을 보여주는 그래프이며, 표 2는 λ^{trans} 값에 따른 데드라인 미스 횟수를 나타낸다. 해상도의 인기도 조합은 비트율당 비디오 품질에는 영향이 있으나, 데드라인 미스 횟수에는 영향이 없었다.

[0095] 표 2: λ^{trans} 값에 따른 데드라인 미스 횟수

기법	λ^{trans}				
	1	2	3	4	5
알고리즘	0	0	0	0	0
RR+HF	9989	86567	129717	173011	216060
WF+HF	5765	8833	12142	14847	17277
RR+LF	0	0	0	0	0
WF+LF	0	0	0	0	0
RR+DF	0	0	0	0	3
WF+DF	1	1	6	11	21
N^{node}	86608	173244	259452	346040	432136

[0096]

[0097] 도 5를 참고하면, 모든 경우에서 실시예에서 제안된 방법이 비트율 당 비디오 품질이 다른 방식보다 훨씬 높다는 것을 알 수 있다. 실시예에서 제안된 방법을 제외했을 때, 비트율 당 비디오 품질이 가장 높은 체계는 RR+HF이며, 비트율 당 비디오 품질이 제안된 알고리즘보다 평균 4.41% 낮은 비디오 품질을 갖는다. 반면, 비트율 당 비디오 품질이 가장 낮은 체계는 RR + LF이며, 비트율 당 평균 비디오 품질이 제안된 알고리즘보다 9.428% 낮음을 확인할 수 있다.

[0098] 데드라인 미스의 측면에서도 실시예에서 제안된 방법의 우수함을 확인할 수 있다. 실시예에서 제안된 방법은 데드라인 미스가 발생하지 않는다. 그러나 RR+LF와 WF+LF를 제외한 다른 기법들은 데드라인 미스가 발생함을 확인할 수 있다. 평균적으로 데드라인 미스가 가장 많은 기법은 RR+HF이며 λ^{trans} 가 1일 때는 11.534%의 데드라인 미스를, λ^{trans} 가 2~5일 때는 거의 절반에 가까운 태스크가 데드라인 미스가 발생함을 확인할 수 있다. RR+DF는 로드가 λ^{trans} 가 1~4일 때는 데드라인 미스가 발생하지 않았으나, λ^{trans} 가 5일 때는 데드라인 미스가 발생함을 확인할 수 있다.

[0099] 데드라인 범위에 따른 결과에 대하여 설명하기로 한다. 태스크의 데드라인의 범위가 비트율 당 비디오 품질 및 데드라인 미스 횟수에 끼치는 영향에 대하여 실험이 수행될 수 있다. 이때, 태스크의 범위를 다양하게 하고, λ^{trans} 를 5로, N^{node} 를 10으로 고정하여 실험이 진행될 수 있다. 도 6은 λ^{trans} 값에 따른 비트율 당 비디오 품질을 보여주는 그래프이며, 표 3은 λ^{trans} 값에 따른 데드라인 미스 횟수를 나타낸다.

[0100] 표 3: 데드라인 범위 값에 따른 데드라인 미스 횟수

기법	λ^{trans}				
	[3, 20]	[3, 30]	[3, 40]	[3, 50]	[3, 60]
알고리즘	0	0	0	0	0
RR+HF	216060	216054	216045	216045	216045
WF+HF	17277	16129	8660	8660	8660
RR+LF	0	0	0	0	0
WF+LF	0	0	0	0	0
RR+DF	3	2	1	1	1
WF+DF	21	25	9	9	9
N^{node}	432136				

[0101]

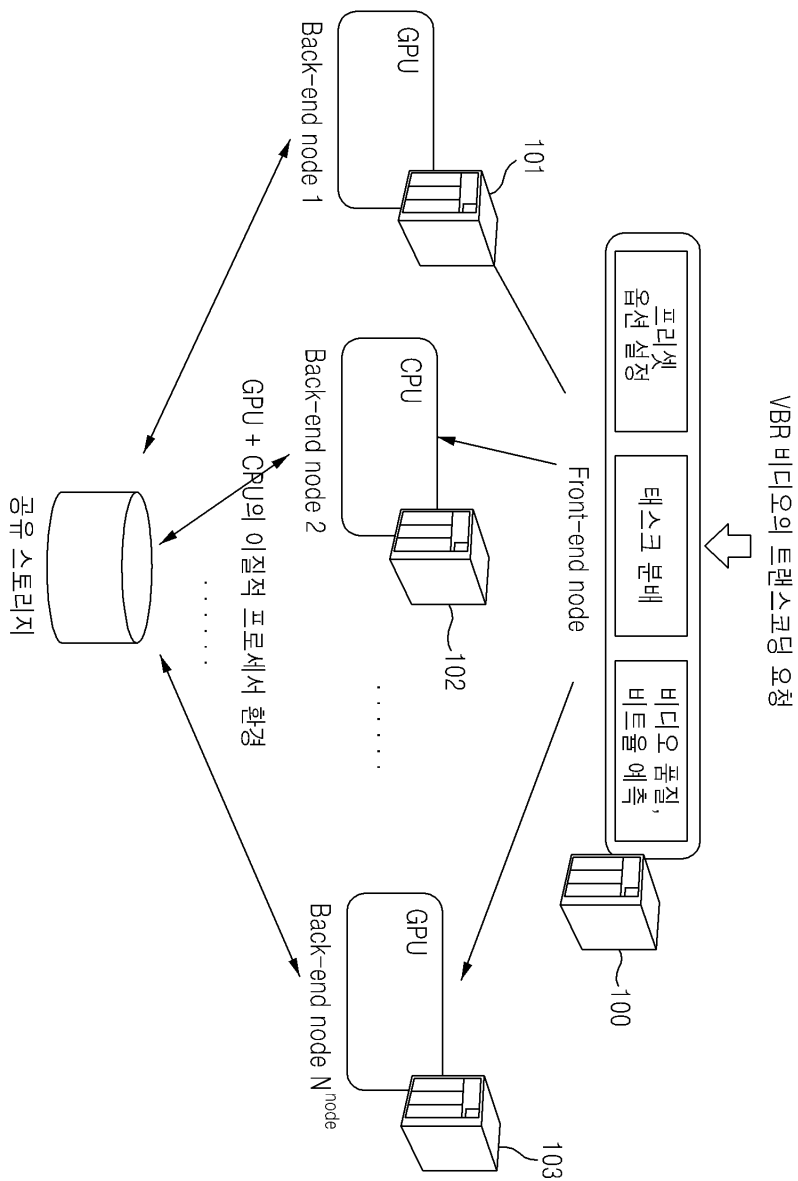
[0102] 이때, 모든 경우에서 실시예에서 제안된 방법이 비트율 당 비디오 품질이 다른 방식보다 훨씬 높다는 것을 확인할 수 있다. 실시예에서 제안된 방법을 제외했을 때, 비트율 당 비디오 품질이 가장 높은 방법은 RR+HF이며, 비트율 당 비디오 품질이 제안된 알고리즘보다 평균 4.404% 낮은 비디오 품질을 갖는 것을 확인할 수 있다. 반면 비트율 당 비디오 품질이 가장 낮은 방법은 RR + LF이며, 비트율 당 평균 비디오 품질이 제안된 알고리즘보

다 9.41% 낮음을 확인할 수 있다.

- [0103] 데드라인 미스 횟수 역시 실시예에서 제안된 방법이 모든 경우에서 우수하다. 실시예에서 제안된 방법과 RR+LF와 WF+LF은 데드라인 미스가 일어나지 않으나, 다른 기법들은 데드라인 미스가 발생하였다. 데드라인 미스가 발생한 모든 기법이 데드라인 범위가 늘어날수록 데드라인 미스가 줄어드는 것을 확인할 수 있다. 평균적으로 데드라인 미스가 가장 많은 기법은 RR+HF이며 모든 데드라인 범위에서 약 절반의 태스크의 데드라인 미스가 발생함을 확인할 수 있다.
- [0104] 이러한 실험들을 통해, 비트율 당 비디오 품질뿐만 아니라 데드라인을 만족시키는 측면에서도 실시예에서 제안된 방법이 언제나 다른 기법보다 우수하다는 것이 입증될 수 있다.
- [0105] 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, 프로세서, 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.
- [0106] 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.
- [0107] 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.
- [0108] 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.
- [0109] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

도면1



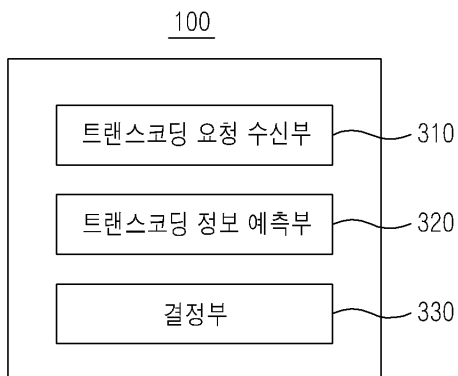
도면2

```

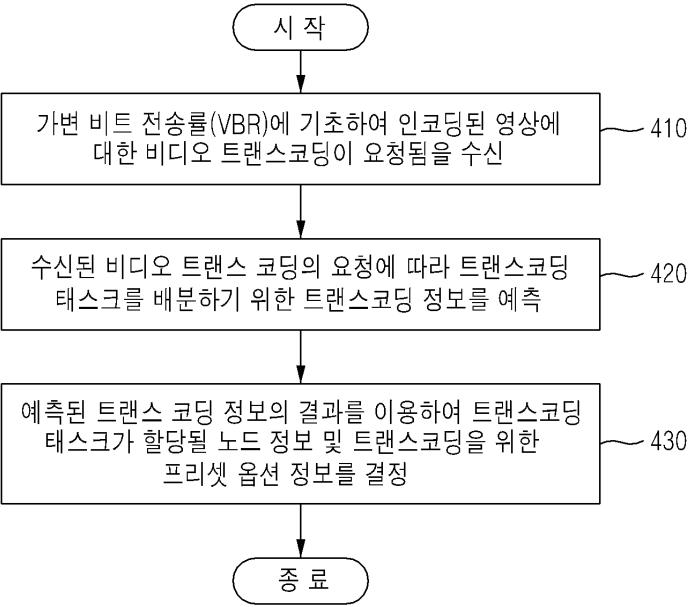
1: Temporary variables:  $\forall i, (i = 1, \dots, N^{\text{task}}), X_i^{\text{temp}}$  and  $Y_i^{\text{temp}}$ ;
2: Array of all the values of  $U_{i,j,k} : A$ ;
3: for  $i = 1$  to  $N^{\text{task}}$  do
4:    $X_i^{\text{temp}} \leftarrow X_i^{\text{low}}$  ;
5:    $Y_i^{\text{temp}} \leftarrow Y_i^{\text{low}}$  ;
6: end for
7: while  $A \neq \emptyset$  do
8:   Choose the highest value of  $U_{| \text{task}, | \text{node}, | \text{para}}$  from  $A$ ;
9:   if  $G_{| \text{task}, | \text{node}, | \text{para}} < G_{| \text{task}, | \text{node}, | \text{para}}$  then
10:    if  $\forall \tau_i \in A_{| \text{node}}, \sum \tau_m \in L_{i, | \text{node}} T_m, X_m, Y_m \leq D_m$  then
11:       $X_{| \text{task}}^{\text{temp}} \leftarrow | \text{node}$  ;
12:       $Y_{| \text{task}}^{\text{temp}} \leftarrow | \text{para}$  ;
13:    end if
14:  end if
15:  Removes  $U_{| \text{task}, | \text{node}, | \text{para}}$  from  $A$ 
16: end while
17: for  $i = 1$  to  $N^{\text{task}}$  do
18:    $X_i \leftarrow X_i^{\text{temp}}$  ;
19:    $Y_i \leftarrow Y_i^{\text{temp}}$  ;
20: end for

```

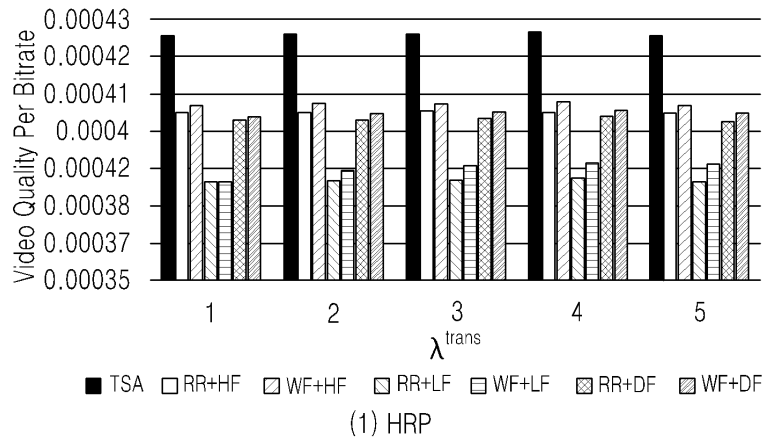
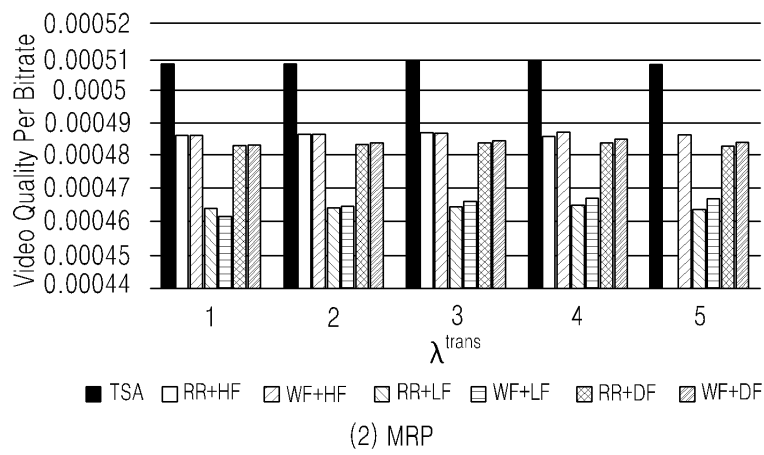
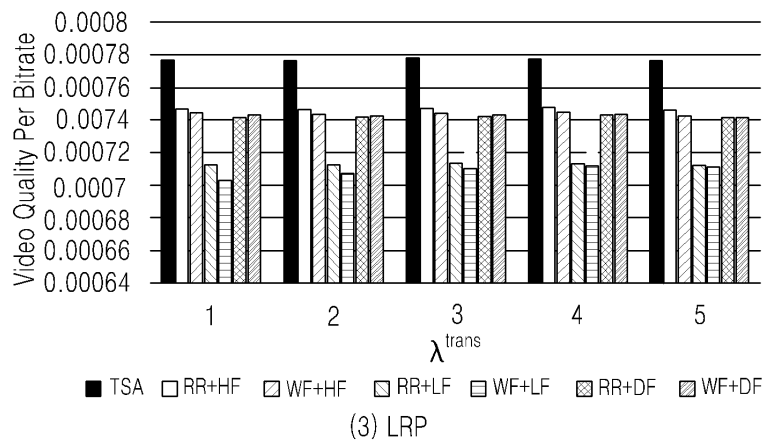
도면3



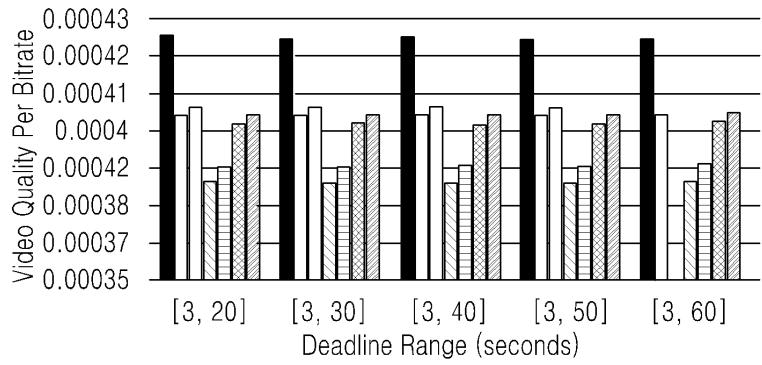
도면4



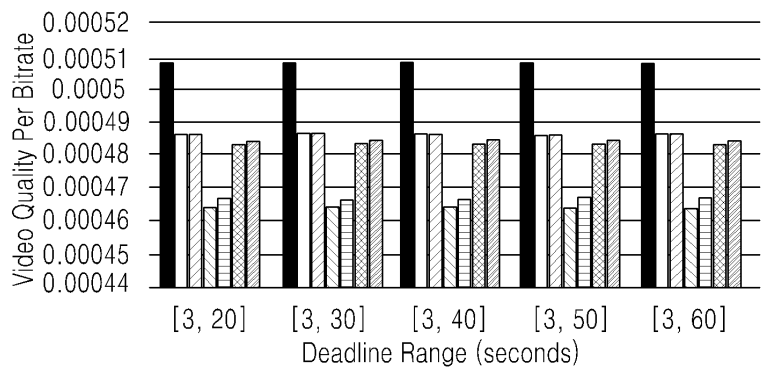
도면5



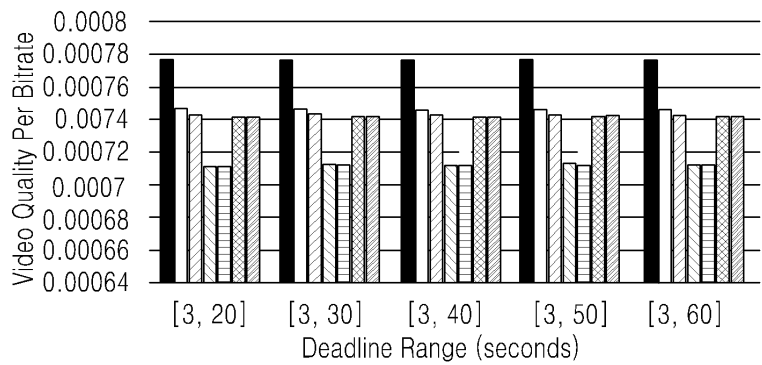
도면6



(1) HRP



(2) MRP



(3) LRP