



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년08월13일
(11) 등록번호 10-2010692
(24) 등록일자 2019년08월07일

(51) 국제특허분류(Int. Cl.)
H04N 21/2343 (2011.01) G06F 1/32 (2019.01)
(52) CPC특허분류
H04N 21/234309 (2013.01)
G06F 1/324 (2019.01)
(21) 출원번호 10-2018-0016081
(22) 출원일자 2018년02월09일
심사청구일자 2018년02월09일
(56) 선행기술조사문헌
KR101411411 B1*
Minseok Song 외 2명, "Scheduling a Video Transcoding Server to Save Energy", CM Transactions on Multimedia Computing Communications and Applications, vol 11. no. 2s, Article 45,(2015.02.) 1부.*
KR1020160099699 A
KR101468357 B1
*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)
김규환
경기도 시흥시 인선길 77, 241동 501호(장곡동, 매희마을삼성아파트)
이정우
인천광역시 남동구 문화서로65번길 28-13, 1층(구월동)
(74) 대리인
양성보

전체 청구항 수 : 총 5 항

심사관 : 홍기완

(54) 발명의 명칭 트랜스코딩 서버의 소모 전력을 제한하면서 트랜스코딩 처리율을 최대화하는 방법

(57) 요약

일 실시예에 따른 트랜스코딩 서버에서 수행되는 트랜스코딩 처리 방법은, 트랜스코딩의 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 단계; 에너지 한계가 입력됨에 따라 상기 모델링된 트랜스코딩 처리율로부터 상기 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택하는 단계; 및 상기 선택된 각각의 CPU의 주파수 레벨에 기초하여 상기 트랜스코딩 요청이 처리됨에 따라 트랜스코딩 전력을 소모하는 단계를 포함할 수 있다.

대표도 - 도1

```

1: Temporary variables:  $i, m$  and  $k$ ;
2: for  $i = 0$  to  $\frac{E_{\text{limit}}}{L}$  do
3:   for  $m = 0$  to  $N^{\text{cpu}}$  do
4:      $C_{i,m}^{\text{trans}} \leftarrow 0$ ;
5:   end for
6: end for
7: for  $i = 1$  to  $\frac{E_{\text{limit}}}{L}$  do
8:   for  $m = 1$  to  $N^{\text{cpu}}$  do
9:     for  $k = 1$  to  $N^{\text{freq}}$  do
10:      if  $i - P_{m,k} \geq 0$  then
11:        if  $C_{i,m}^{\text{trans}} < C_{i-P_{m,k},m-1}^{\text{trans}} + C_m^{\text{trans}}(f_m(k))$  then
12:           $C_{i,m}^{\text{trans}} \leftarrow C_{i-P_{m,k},m-1}^{\text{trans}} + C_m^{\text{trans}}(f_m(k))$ ;
13:           $C_{i,m}^{\text{back}} \leftarrow k$ ;
14:        end if
15:      end if
16:    end for
17:  end for
18: end for
19:  $i \leftarrow \frac{E_{\text{limit}}}{L}$ ;
20:  $m \leftarrow N^{\text{cpu}}$ ;
21: while  $m > 0$  do
22:    $F_m \leftarrow C_{i,m}^{\text{back}}$ ;
23:    $i \leftarrow i - P_{m,F_m}$ ;
24:    $m \leftarrow m - 1$ ;
25: end while

```

(52) CPC특허분류

H04N 21/23439 (2013.01)

H04N 21/4436 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 NRF-2017M3C4A7080248

부처명 과학기술정보통신부

연구관리전문기관 한국연구재단

연구사업명 원천기술개발사업(차세대정보컴퓨팅기술개발사업)

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

기 여 율 1/1

주관기관 인하대학교

연구기간 2017.11.01 ~ 2018.02.28

명세서

청구범위

청구항 1

트랜스코딩 서버에서 수행되는 트랜스코딩 처리 방법에 있어서,

트랜스코딩 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 단계;

에너지 한계가 입력됨에 따라 상기 모델링된 트랜스코딩 처리율로부터 상기 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택하는 단계; 및

상기 선택된 각각의 CPU의 주파수 레벨에 기초하여 상기 트랜스코딩 요청이 처리됨에 따라 트랜스코딩 전력을 소모하는 단계

를 포함하고,

상기 에너지 한계가 입력됨에 따라 상기 모델링된 트랜스코딩 처리율로부터 상기 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택하는 단계는,

첫번째 CPU부터 마지막 CPU를 포함하는 상기 서로 다른 종류로 구성된 적어도 하나 이상의 CPU에 대하여 각각의

CPU에서 사용된 에너지에 기초하여 획득되는 최대의 트랜스코딩 처리율($C_{i,m}^{trans}$)을 계산하는 초기화 과정, 상기 초기화 과정을 수행함에 따라 상기 최대의 트랜스코딩 처리율을 점화식

$$C_{i,m}^{trans} = \max \left(\max_{m=1, \dots, N_m^{freq}} (C_{i-1,m-P_m, F_m}^{trans} + C^{trans}(f)) \right)$$

()으로 계산하는 점화식 수립 과정 및 상기 최대의 트랜스코딩 처리율에서 상기 마지막 CPU로부터 상기 첫번째 CPU까지의 주파수 레벨에 대한 역추적을 통하여 각각의 CPU에서 사용되는 CPU의 주파수 레벨을 저장하는 역추적 과정을 수행하는 단계

를 포함하는 트랜스코딩 처리 방법.

청구항 2

제1항에 있어서,

상기 트랜스코딩의 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 단계는,

상기 트랜스코딩이 처리되는 트랜스코딩 시간이 주파수 레벨에 반비례하고, 상기 트랜스코딩 시간에 따라 트랜스코딩 처리율이 판단되는 단계

를 포함하는 트랜스코딩 처리 방법.

청구항 3

삭제

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 트랜스코딩 처리율의 범위는, 상기 트랜스코딩이 처리되는 모든 CPU에서 주파수의 레벨 중 가장 낮은 주파수의 레벨이 선택되었을 때의 처리율로부터 상기 모든 CPU에서 주파수 레벨 중 기 설정된 기준 이상의 주파수의

레벨이 선택되었을 때의 처리율의 범위를 포함하는 트랜스코딩 처리 방법.

청구항 6

제1항에 있어서,

상기 트랜스코딩의 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 단계는,

트랜스코딩 세션과 스트리밍 세션으로 구성된 부문형 비디오에서 상기 트랜스코딩 세션에 비디오 세그먼트들이 트랜스코딩되어 저장되고, 클라이언트로부터 비디오 세그먼트가 요청됨에 응답하여 상기 트랜스코딩되어 저장된 비디오 세그먼트들을 스트리밍 세션에 전달하고, 상기 스트리밍 세션에서 상기 트랜스코딩되어 저장된 비디오 세그먼트들을 상기 클라이언트에게 전달하는 단계

를 포함하는 트랜스코딩 처리 방법.

청구항 7

트랜스코딩 서버에 있어서,

트랜스코딩 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 모델링부;

에너지 한계가 입력됨에 따라 상기 모델링된 트랜스코딩 처리율로부터 상기 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택하는 주파수 선택부; 및

상기 선택된 각각의 CPU의 주파수 레벨에 기초하여 상기 트랜스코딩 요청이 처리됨에 따라 트랜스코딩 전력을 소모하는 전력 소모부

를 포함하고,

상기 주파수 선택부는,

첫번째 CPU부터 마지막 CPU를 포함하는 상기 서로 다른 종류로 구성된 적어도 하나 이상의 CPU에 대하여 각각의

CPU에서 사용된 에너지에 기초하여 획득되는 최대의 트랜스코딩 처리율($C_{i,m}^{trans}$)을 계산하는 초기화 과정, 상기 초기화 과정을 수행함에 따라 상기 최대의 트랜스코딩 처리율을 점화식

(
$$C_{i,m}^{trans} = \max (\max_{m=1, \dots, N_m^{freq}} (C_{i-1,m-P_m, F_m}^{trans} + C^{trans}(f)))$$
)으로 계산하는 점화식 수립 과정 및 상기 최대의 트랜스코딩 처리율에서 상기 마지막 CPU로부터 상기 첫번째 CPU까지의 주파수 레벨에 대한 역추적을 통하여 각각의 CPU에서 사용되는 CPU의 주파수 레벨을 저장하는 역추적 과정을 수행하는

트랜스코딩 처리 서버.

발명의 설명

기술 분야

[0001] 아래의 설명은 트랜스코딩 서버의 소모 전력을 제한하면서 트랜스코딩 처리율을 최대화하는 기술에 관한 것이다.

배경 기술

[0002]최근 멀티미디어 및 네트워크 기술의 발전으로 동적 적응 스트리밍(Dynamically Adaptive Streaming) 기술이 보편화되었다. 동적 적응 스트리밍은 1개의 비디오를 여러 비트율, 해상도, 코덱의 다양한 버전으로 변환하여야 하며, 이와 같은 변환 과정을 트랜스코딩이라고 한다.

- [0003] 트랜스코딩 연산은 기본적으로 많은 연산을 요구하며, 이에 따라서 높은 CPU 소모 전력을 요구한다. 또한 높은 트랜스코딩 처리 요구량에 대처하기 위해서 1개의 CPU 만으로 트랜스코딩 처리는 불가능하며, 여러 CPU들이 협업을 통해서 트랜스코딩을 처리하여야 한다. 지속적으로 증가하는 트랜스코딩 요구량에 대처하기 위해서는 트랜스코딩 서버에 새로운 CPU를 지속적으로 추가하여야 하며, 이질적인 속성을 갖은 많은 CPU가 하나의 트랜스코딩 서버를 구성하게 된다.
- [0004] CPU 전력 소비를 줄이는 효과적인 방법은 프로세서 주파수에 따른 전압을 동시에 변화시키는 동적 전압 주파수 조절(DVFS: Dynamic Voltage and Frequency Scaling) 기술을 사용하는 것이다. CPU는 여러 개의 주파수를 갖게 되며, 주파수를 낮출수록 인가되는 전압이 떨어지므로, 전력 소모를 효과적으로 줄일 수 있으나, 트랜스코딩 시간을 증가시킴으로써, 이를 고려한 효과적인 주파수 선택 기법이 필수적이다. 특히, 주파수가 이질적인 CPU로 구성된 서버에서 어떻게 주파수를 선택하느냐에 따라서 전력 소모량은 큰 차이를 보인다.
- [0005] 참고자료: KR 10-1468357

발명의 내용

해결하려는 과제

- [0006] 본 발명은 다양한 주파수를 가지는 이질적 CPU가 포함된 트랜스코딩 서버에서 트랜스코딩 에너지 소모량을 제한하면서, 트랜스코딩 처리율을 최대화 하는 CPU 주파수 선택 기법을 제공할 수 있다.
- [0007] 보다 구체적으로, 트랜스코딩 처리율을 CPU 주파수에 근거하여 모델링하고, 에너지 한계가 주어졌을 때, 트랜스코딩 처리율을 최대화하는 문제를 제안하며, 제안된 문제를 풀기 위해서 동적 프로그래밍 기법을 활용하여 각 CPU의 주파수를 선택하는 최적화 방법을 제안한다.

과제의 해결 수단

- [0008] 트랜스코딩 서버에서 수행되는 트랜스코딩 처리 방법은, 트랜스코딩 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 단계; 에너지 한계가 입력됨에 따라 상기 모델링된 트랜스코딩 처리율로부터 상기 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택하는 단계; 및 상기 선택된 각각의 CPU의 주파수 레벨에 기초하여 상기 트랜스코딩 요청이 처리됨에 따라 트랜스코딩 전력을 소모하는 단계를 포함할 수 있다.
- [0009] 상기 트랜스코딩의 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 단계는, 상기 트랜스코딩이 처리되는 트랜스코딩 시간이 주파수 레벨에 반비례하고, 상기 트랜스코딩 시간에 따라 트랜스코딩 처리율이 판단되는 단계를 포함하고, 각각의 CPU로부터 복수 개의 서로 다른 주파수 레벨에서 실행될 수 있다.
- [0010] 상기 에너지 한계가 입력됨에 따라 상기 모델링된 트랜스코딩 처리율로부터 상기 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택하는 단계는, 첫번째 CPU부터 마지막 CPU를 포함하는 상기 서로 다른 종류로 구성된 적어도 하나 이상의 CPU에 대하여 각각의 CPU에서 사용된 에너지에 기초하여 획득되는 최대의 트랜스코딩 처리율을 계산하는 단계를 포함할 수 있다.
- [0011] 상기 에너지 한계가 입력됨에 따라 상기 모델링된 트랜스코딩 처리율로부터 상기 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택하는 단계는, 상기 최대의 트랜스코딩 처리율에서 상기 마지막 CPU로부터 상기 첫번째 CPU까지의 주파수 레벨에 대한 역추적을 통하여 각각의 CPU에서 사용되는 CPU의 주파수 레벨을 저장하는 단계를 포함할 수 있다.
- [0012] 상기 트랜스코딩 처리율의 범위는, 상기 트랜스코딩이 처리되는 모든 CPU에서 주파수의 레벨 중 가장 낮은 주파수의 레벨이 선택되었을 때의 처리율부터 상기 모든 CPU에서 주파수 레벨 중 기 설정된 기준 이상의 주파수의 레벨이 선택되었을 때의 처리율의 범위를 포함할 수 있다.
- [0013] 상기 트랜스코딩의 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 단계는, 트랜스코딩 세션과 스트리밍 세션으로 구성된 부문형 비디오에서 상기 트랜스코딩 세션에

비디오 세그먼트들이 트랜스코딩되어 저장되고, 클라이언트로부터 비디오 세그먼트가 요청됨에 응답하여 상기 트랜스코딩되어 저장된 비디오 세그먼트들을 스트리밍 세션에 전달하고, 상기 스트리밍 세션에서 상기 트랜스코딩되어 저장된 비디오 세그먼트들을 상기 클라이언트에게 전달하는 단계를 포함할 수 있다.

[0014] 트랜스코딩 서버는, 트랜스코딩 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링하는 모델링부; 에너지 한계가 입력됨에 따라 상기 모델링된 트랜스코딩 처리율로부터 상기 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택하는 주파수 선택부; 및 상기 선택된 각각의 CPU의 주파수 레벨에 기초하여 상기 트랜스코딩 요청이 처리됨에 따라 트랜스코딩 전력을 소모하는 전력 소모부를 포함할 수 있다.

발명의 효과

[0015] 본 발명은 동일한 서버 환경에서 CPU 주파수 선택 정책에 따라 트랜스코딩 소모 전력을 효과적으로 감소시킬 수 있다.

[0016] 본 발명은 전력/에너지 소비량을 제한하고, 트랜스코딩 처리율을 최대화함으로써 주어진 에너지를 효과적으로 사용할 수 있다.

도면의 간단한 설명

[0017] 도 1은 일 실시예에 따른 트랜스코딩 서버에서 수행되는 주파수 선택 알고리즘의 수도코드를 나타낸 예이다.
 도 2는 일 실시예에 따른 트랜스코딩 서버의 구성을 설명하기 위한 블록도이다.
 도 3은 일 실시예에 따른 트랜스코딩 서버의 트랜스코딩 방법을 설명하기 위한 흐름도이다.
 도 4는 일 실시예에 따른 트랜스코딩 서버에서 CPU 주파수와 측정 전력 값과의 관계를 설명하기 위한 도면이다.
 도 5는 일 실시예에 따른 트랜스코딩 서버에서 에너지 제한 비율에 따른 트랜스코딩 처리율을 나타낸 그래프이다.
 도 6은 일 실시예에 따른 트랜스코딩 서버에서 제안한 FSA와 UA 알고리즘의 트랜스코딩 처리량을 비교한 그래프이다.
 도 7은 일 실시예에 따른 트랜스코딩 서버에서 에너지 한계를 고정할 때의 트랜스코딩 세션 시간에 따른 처리율을 나타낸 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0018] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0019] 아래의 실시예에서는 이질적 주파수의 CPU로 구성된 트랜스코딩 서버에서 에너지 한도가 주어졌을 때의 트랜스코딩 처리율을 최대화하는 기법에 관한 것이다. 트랜스코딩 처리율을 CPU 선택 주파수에 따라 큰 차이를 보이며, CPU 주파수는 소모 전력에 큰 영향을 끼친다. 이에 따라 트랜스코딩 처리율을 CPU 주파수에 근거하여 모델링하고, 에너지 한도가 주어졌을 때, 트랜스코딩 처리율을 최대화하기 위하여 동적 프로그래밍 기법을 활용하여 각 CPU에 인가되는 주파수를 선택하는 최적의 방법을 제안한다. 실시예에 따르면 상기 제안된 기법이 전력/에너지 소비량을 제한하고, 트랜스코딩 처리율을 최대화함으로써 주어진 에너지를 효과적으로 사용함을 확인할 수 있다.

[0020] 도 1은 일 실시예에 따른 트랜스코딩 서버에서 수행되는 주파수 선택 알고리즘의 수도코드를 나타낸 예이다.

[0021] CPU의 주파수 선택 문제를 해결하기 위하여 일례로 동적 프로그래밍 기법이 활용될 수 있다. 트랜스코딩 서버에서 최적의 주파수를 선택하기 위한 주파수 선택 알고리즘이 제안될 수 있다. 이때, 편의상 제안하는 동적 프로그래밍 알고리즘을 주파수 선택 알고리즘(FSA: frequency selection algorithm)이라고 기재하기로 한다.

[0022] 주파수 선택 알고리즘은 초기화 단계, 점화식 수립 단계, 역추적 단계로 구성될 수 있다.

[0023] 초기화 단계에서, $C_{i,m}^{trans}$ 는 i (i 는 자연수)개의 CPU에 대하여 첫번째 CPU부터 마지막 CPU까지의 모든 CPU에 대하여 에너지를 일정량(예를 들면, m)만큼 사용하였을 때, 획득할 수 있는 최대의 트랜스코딩 처리율을 의미한다. 이때, $\forall i, m, C_{i,m}^{trans}$ 을 모두 0으로 초기화시킬 수 있다.

[0024] 점화식 수립 단계에서, $C_{i,m}^{trans}$ 는 다음과 같은 점화식으로 계산될 수 있다. 만약 $C_{i-1,m-P_m, F_m}^{trans} > 0$ 이면, $C_{i,m}^{trans}, (i = 2, \dots, N_{cpu})$ 는 다음의 점화식으로 갱신될 수 있다.

$$C_{i,m}^{trans} = \max \left(\max_{m=1, \dots, N_m^{freq}} (C_{i-1,m-P_m, F_m}^{trans} + C^{trans}(f)) \right)$$

[0025]

[0026] 역추적 단계에서, $C_{i,m}^{trans}$ 는 i (i 는 자연수)개의 CPU에 대하여 첫번째 CPU부터 마지막 CPU까지의 모든 CPU에 대한 최대의 트랜스코딩 처리율을 의미한다. 이때, 마지막 CPU의 N_{cpu} , 에너지 제한 비율 $\frac{E^{limit}}{L}$ 로부터 시작하여 역으로 첫번째 CPU의 테이블을 추적하여 각각의 CPU에 사용될 주파수 레벨 f 를 탐색할 수 있다. 이를 위하여 변수 $C_{i,m}^{back}$ 는 $C_{i,m}^{trans}$ 때의 각각의 CPU의 주파수 레벨을 저장할 수 있다. 이와 같이, 역추적을 통하여 각각의 CPU의 주파수 레벨을 찾을 수 있다.

[0027] 도 2는 일 실시예에 따른 트랜스코딩 서버의 구성을 설명하기 위한 블록도이고, 도 3은 일 실시예에 따른 트랜스코딩 서버의 트랜스코딩 방법을 설명하기 위한 흐름도이다.

[0028] 트랜스코딩 서버(100)는 모델링부(210), 주파수 선택부(220) 및 전력 소모부(230)를 포함할 수 있다. 이러한 구성요소들은 트랜스코딩 서버(100)에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 구성요소들은 도 3의 트랜스코딩 방법이 포함하는 단계들(310 내지 330)을 수행하도록 트랜스코딩 서버(100)를 제어할 수 있다. 이때, 구성요소들은 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다.

[0029] 프로세서는 트랜스코딩 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 서버를 제어할 수 있다. 이때, 프로세서 및 프로세서가 포함하는 모델링부(210), 주파수 선택부(220) 및 전력 소모부(230) 각각은 메모리에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계들(310 내지 330)을 실행하기 위한 프로세서의 서로 다른 기능적 표현들일 수 있다.

[0030] 단계(310)에서 모델링부(210)는 트랜스코딩 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링할 수 있다. 모델링부(210)는 트랜스코딩 세션과 스트리밍 세션으로 구성된 부분형 비디오에서 트랜스코딩 세션에 비디오 세그먼트들이 트랜스코딩되어 저장되고, 클라이언트로부터 비디오 세그먼트가 요청됨에 응답하여 트랜스코딩되어 저장된 비디오 세그먼트들을 스트리밍 세션에 전달하고, 스트리밍 세션에서 트랜스코딩되어 저장된 비디오 세그먼트들을 클라이언트에게 전달할 수 있다. 모델링부(210)는 트랜스코딩이 처리되는 트랜스코딩 시간이 주파수 레벨에 반비례하고, 트랜스코딩 시간에 따라 트랜스코딩 처리율을 판단할 수 있다.

[0031] 단계(320)에서 주파수 선택부(220)는 에너지 한계가 입력됨에 따라 모델링된 트랜스코딩 처리율로부터 에너지 한계 내에서 각각의 CPU의 주파수 레벨을 선택할 수 있다. 이때, 트랜스코딩 처리율의 범위는, 트랜스코딩이 처리되는 모든 CPU에서 주파수의 레벨 중 가장 낮은 주파수의 레벨이 선택되었을 때의 처리율부터 모든 CPU에서 주파수 레벨 중 기 설정된 기준 이상의 주파수의 레벨이 선택되었을 때의 처리율의 범위를 포함할 수 있다. 주파수 선택부(220)는 첫번째 CPU부터 마지막 CPU를 포함하는 서로 다른 종류로 구성된 적어도 하나 이상의 CPU에

대하여 각각의 CPU에서 사용된 에너지에 기초하여 획득되는 최대의 트랜스코딩 처리율을 계산할 수 있다. 주파수 선택부(220)는 최대의 트랜스코딩 처리율에서 마지막 CPU로부터 첫번째 CPU까지의 주파수 레벨에 대한 역추적을 통하여 각각의 CPU에서 사용되는 CPU의 주파수 레벨을 저장할 수 있다.

[0032] 단계(330)에서 전력 소모부(230)는 선택된 각각의 CPU의 주파수 레벨에 기초하여 트랜스코딩 요청이 처리됨에 따라 트랜스코딩 전력을 소모할 수 있다.

[0033] 도 4는 일 실시예에 따른 트랜스코딩 서버에서 CPU 주파수와 측정 전력 값과의 관계를 설명하기 위한 도면이다.

[0034] 트랜스코딩 서버에서 트랜스코딩 요청에 기반하여 서로 다른 종류로 구성된 적어도 하나 이상의 CPU의 트랜스코딩 처리율을 모델링할 수 있다. 이때, 주문형 비디오 시스템에서 트랜스코딩 세션 및 스트리밍 세션으로 구성될 수 있다. 트랜스코딩 세션은 L의 시간동안 지속되며 비디오 세그먼트들이 트랜스코딩되어 저장 장치에 저장된 후, 스트리밍 세션에 해당 비디오 세그먼트들을 요청한 클라이언트에게 전달될 수 있다. 다시 말해서, 트랜스코딩 세션에 비디오 세그먼트들이 트랜스코딩되어 저장되고, 클라이언트로부터 비디오 세그먼트가 요청됨에 응답하여 트랜스코딩되어 저장된 비디오 세그먼트들을 스트리밍 세션에 전달하고, 스트리밍 세션을 통하여 클라이언트로부터 요청된 비디오 세그먼트에 대응하는 트랜스코딩되어 저장된 비디오 세그먼트들을 클라이언트에게 전달할 수 있다.

[0035] 트랜스코딩의 요청이 증가함에 따라 서비스 공급자는 새로운 프로세서를 추가해야 한다. 이에 따라 복수의 서로 다른 종류들로 이루어진 CPU들이 구성될 수 있다. N^{cpu} 를 CPU의 개수라고 하자. 각 CPU가 여러 가지 개별 주파수 레벨에서 실행될 수 있다고 가정하면, $f_m(n)$ 은 m(m은 자연수)번째 CPU의 n(n은 자연수)번째 주파수 레벨을 의미한다. N_m^{freq} 를 m번째 CPU의 주파수의 개수라고 하자. 기본 주파수(일반적인 주파수)를 f_{base} 라고 할 때, 기본 주파수는 트랜스코딩 서버에서 동작하는 CPU 중 가장 높은 주파수를 의미한다. 기본 주파수 f_{base} 는 아래의 수식 1과 같이 정의된다.

[0036] 수식 1:

$$f_{base} = \max_{m=1, \dots, N^{cpu}} f_m(N_m^{freq})$$

[0037]

[0038] 그리고, 트랜스코딩 시간은 주파수 레벨에 반비례한다. 예를 들면, 주파수 레벨이 f_{base} 에서 $\frac{1}{2}f_{base}$ 가 되면 트랜스코딩 시간이 2배로 증가하기 때문에 동일한 시간에 트랜스코딩 처리율은 1/2이 된다. 이에 따라 주파수 f 에 대한 트랜스코딩 처리율 $C^{trans}(f)$ 은 주파수 f 의 함수로 다음과 같이 정의된다(수식 2).

[0039] 수식 2:

$$C^{trans}(f) = \frac{f}{f_{base}}$$

[0040]

[0041] 트랜스코딩 서버는 트랜스코딩 세션에서의 소모 에너지를 미리 정의된 에너지 한계인 한계값(E^{limit})을 초과하지 않도록 에너지를 제한하면서 트랜스코딩 처리율을 최대로 하기 위하여 각 CPU 주파수를 선택할 수 있다. 이를 위하여 CPU 주파수를 선택하는 문제를 다음과 같이 정의할 수 있다.

[0042] CPU 주파수 문제에서 각 CPU의 주파수 레벨은 트랜스코딩 처리율을 최대화하기 위한 목적으로 정의될 수 있다. 트랜스코딩 처리율의 범위는 모든 CPU에서 가장 낮은 주파수가 선택되었을 때의 전체 처리율인

$\sum_{i=1}^{N^{cpu}} C^{trans}(1)$ 모든 CPU에서 가장 높은 주파수가 선택되었을 때의 전체 처리율인 $\sum_{i=1}^{N^{cpu}} C^{trans}(N_m^{freq})$ 의 범위를 포함한다.

$P_{m,n}$ 을 때, m번째 CPU에서 주파수 레벨 n이 선택되었을 때 소비되는 전력 소모량을 (와트: W)라고 가정한다 (m=1, ..., N^{cpu} and n=1, ..., N_m^{freq}). 그리고 F_m 을 m번째 CPU에서 선택된 주파수 레벨이라고 가정한다(m=1, ..., N^{cpu}).

트랜스코딩 서버에서 트랜스코딩 세션 L 시간 동안 소모되는 에너지 소모량을 E^{limit} 로 제한하면서 총 트랜스코딩 처리율을 최대화하기 위하여 m번째 CPU에서 선택되는 주파수 레벨 F_m 값을 획득하는 최적화 문제를 CPU 주파수 선택 문제라고 정의할 수 있다. 이러한 CPU 주파수 선택 문제는 다음과 같이 정의될 수 있다.

수식 3:

$$\begin{aligned} & \text{Maximize} \quad \sum_{i=1}^{N^{cpu}} \sum_{j=1}^{N_m^{freq}} C^{trans}(f_m(F_m)) \\ & \text{Subject to} \quad \sum_{i=1}^{N^{cpu}} \sum_{j=1}^{N_m^{freq}} P_{m,F_m} L \leq E^{limit} \\ & \quad \quad \quad F_m \in 1, \dots, N_m^{freq} \end{aligned}$$

일례로, N^{cpu} =50일 때의 트랜스코딩 처리율을 시뮬레이션을 수행할 수 있다. 각각의 CPU의 첫번째 주파수는 1.4GHz 부터 1.8GHz 사이에서 임의로 선택될 수 있고, 기 설정된 기준 이상의 주파수(예를 들면, 가장 높은 주파수)는 3.0GHz 부터 3.4GHz 사이에서 임의로 선택되며, 기 설정된 기준에 기반하여 일정 범위의 주파수(예를 들면, 중간 단계 주파수)는 1.8GHz부터 3.0GHz 범위 내에서 임의로 선택될 수 있다. 이와 같이, 선택된 주파수의 전력 값은 도 4에서의 측정값을 활용할 수 있다. 이때, CPU 가 아무런 일을 수행하지 않을 때 아이들 전력(idle power)는 1W부터 40W사이에서 임의로 선택된다고 가정한다.

시뮬레이션을 통하여 에너지 제한 비율 $\frac{E^{limit}}{L}$ 값에 따른 트랜스코딩 처리율을 계산할 수 있다. 모든 CPU에서 기 설정된 기준 이상의 주파수(예를 들면, 최대 주파수)로 선택되었을 때의 전력 값을 1이라고 하였을 때, 모든 CPU에서 최대 주파수로 선택될 때 대비한 소비 전력 비율을 r이라고 하자. 도 5를 참고하면, r에 대한 트랜스코딩 처리율을 보여 주며, r에 비례해서 트랜스코딩 처리율이 증가함을 확인할 수 있다 ($0.5 \leq r \leq 1$).

또한, 동적 프로그래밍 기법의 효율성을 검증하기 위해서 또 다른 알고리즘(편의상, 새로운 알고리즘을 uniform algorithm(UA)라고 함)을 구현할 수 있다. UA 기법은 1)각각의 CPU 주파수 레벨을 1로 초기화하고, 2)에너지 한계를 만족할 때까지 각각 CPU의 주파수를 하나씩 증가시키는 기법이다. 도 6을 참고하면, r에 대한 트랜스코딩 처리율을 보여 주며, UA의 트랜스코딩 처리율이 FSA 기법보다 최대 6.4% 낮은 것을 확인할 수 있다.

모든 CPU에서 최대 주파수로 트랜스코딩 서버를 운용할 경우에는 트랜스코딩 처리가 빨리 끝나게 된다. 이에 비하여, 기 설정된 기준 이하의 주파수(예를 들면, 낮은 주파수)를 운용 시에는 트랜스코딩 시간이 길어지나,

에너지 소모량은 감소하게 된다. 모든 CPU에서 최대의 주파수로 수행 시의 트랜스코딩 시간을 L^* 이라고 하자.

에너지 한계 E^{limit} 을 고정하고, 트랜스코딩 세션 시간 L 을 증가시키면서 FSA 알고리즘을 수행하고, 모든 CPU 에서 최대 주파수로 수행할 때 대비해서 트랜스코딩 처리율을 계산한 결과가 도 7과 같이 나타날 수 있다. 도 7을 참고하면, E^{limit} 을 고정하고 트랜스코딩 세션 L 을 증가 시에는, 낮은 CPU 주파수로 트랜스코딩 서버를 운용할 수 있으므로, 소모 전력을 감소시킬 수 있다. 따라서, 동일 에너지 사용 시 트랜스코딩 처리율이 증가됨을 확인할 수 있다.

[0051] 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.

[0052] 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.

[0053] 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.

[0054] 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.

[0055] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

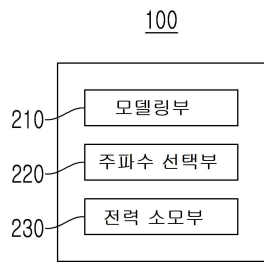
도면1

```

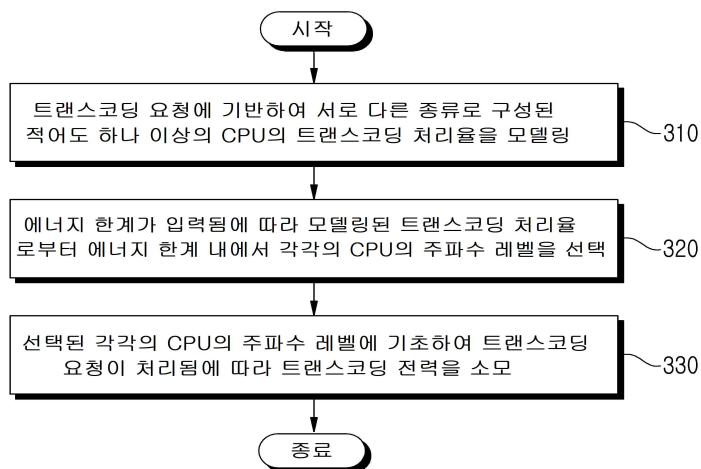
1: Temporary variables:  $i$ ,  $m$  and  $k$ ;
2: for  $i = 0$  to  $\frac{E_{limit}}{L}$  do
3:   for  $m = 0$  to  $N^{cpu}$  do
4:      $C_{i,m}^{trans} \leftarrow 0$ ;
5:   end for
6: end for
7: for  $i = 1$  to  $\frac{E_{limit}}{L}$  do
8:   for  $m = 1$  to  $N_m^{freq}$  do
9:     for  $k = 1$  to  $N_m^{freq}$  do
10:      if  $i - P_{m,k} \geq 0$  then
11:        if  $C_{i,m}^{trans} < C_{i-P_{m,k},m-1}^{trans} + C^{trans}(f_m(k))$  then
12:           $C_{i,m}^{trans} \leftarrow C_{i-P_{m,k},m-1}^{trans} + C^{trans}(f_m(k))$ ;
13:           $C_{i,m}^{back} \leftarrow k$ ;
14:        end if
15:      end if
16:    end for
17:  end for
18: end for
19:  $i \leftarrow \frac{E_{limit}}{L}$ ;
20:  $m \leftarrow N^{cpu}$ ;
21: while  $m > 0$  do
22:    $F_m \leftarrow C_{i,m}^{back}$ ;
23:    $i \leftarrow i - P_{m,F_m}$ ;
24:    $m \leftarrow m - 1$ ;
25: end while

```

도면2



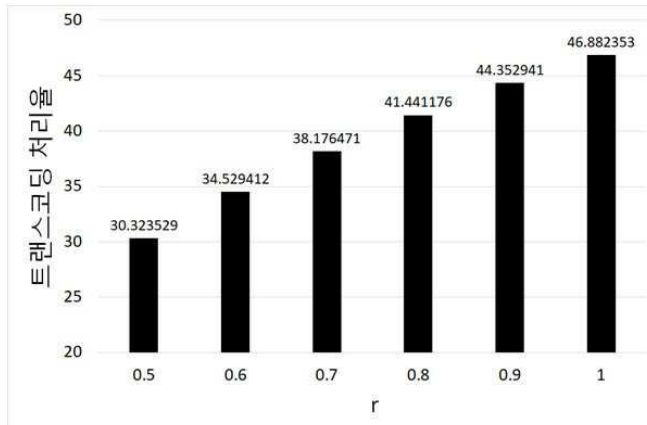
도면3



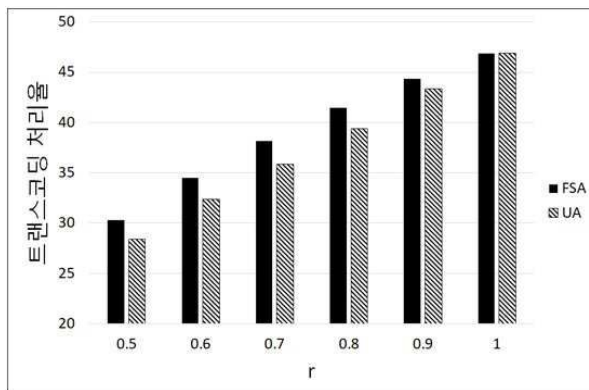
도면4

Table VI. CPU Frequency and Active Power of the CPUs in Our Prototype															
Frequency level	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Frequency (GHz)															
Node 1	1.6	1.7	1.9	2.0	2.1	2.2	2.4	2.5	2.6	2.8	2.9	3.0	3.1	3.3	3.4
Node 2	1.6	1.7	1.8	1.9	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.9	3.0	3.1	3.2
Node 3	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0	3.2	3.4					
Active system-wide power consumption measured (W)															
Node 1	60	61	64	65	66	67	70	71	73	77	80	82	84	90	93
Node 2	51	52	53	54	57	58	59	61	62	64	66	70	72	76	78
Node 3	83	86	91	96	101	107	114	122	128	136					

도면5



도면6



도면7

