



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0068395
(43) 공개일자 2013년06월26일

(51) 국제특허분류(Int. Cl.)

G06F 1/32 (2006.01) G06F 9/46 (2006.01)

(21) 출원번호 10-2011-0135518

(22) 출원일자 2011년12월15일

심사청구일자 2011년12월15일

(71) 출원인

경상대학교산학협력단

경상남도 진주시 진주대로 501 (가좌동)

(72) 발명자

이완연

강원도 춘천시 칠전동 대우아파트 205동 602호

김경훈

경상남도 진주시 상대2동 301-94번지 202호

(74) 대리인

조흥오

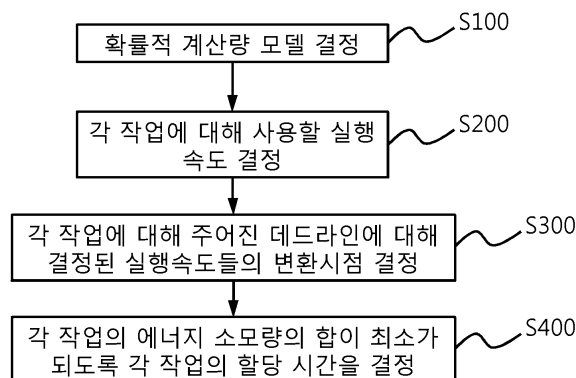
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법

(57) 요약

본 발명은 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법에 관한 것으로서, M개의 실시간 작업(T^m) 각각에 대하여 각 작업의 데드라인(D^m) 이전에 수행해야 할 계산량을 예측하기 위한 확률적 계산량 모델(Φ_c^m)을 결정하는 제1 단계; 각각의 실시간 작업(T^m)에 대해 상기 확률적 계산량 모델(Φ_c^m)에 의한 에너지 소모량을 추정하기 위하여 사용할 실행 속도를 결정하는 제2 단계; 각각의 실시간 작업(T^m)에 대하여 주어진 데드라인(D^m)에 대해 상기 결정된 실행 속도들의 변환 시점을 결정하는 제3 단계; 및 상기 결정된 변환 시점에 기초하여 각각의 실시간 작업(T^m)들의 에너지 소모량의 합이 최소값을 가지도록 각 실시간 작업(T^m)의 할당 시간(A^m)을 결정하는 제4 단계를 포함하는 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법을 제공한다.

대표도 - 도1



특허청구의 범위

청구항 1

변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법에 있어서,

M개의 실시간 작업(T^m) 각각에 대하여 각 작업의 데드라인(D^m) 이전에 수행해야 할 계산량을 예측하기 위한 확률적 계산량 모델(Φ_c^m)을 결정하는 제1 단계;

각각의 실시간 작업(T^m)에 대해 상기 확률적 계산량 모델(Φ_c^m)에 의한 에너지 소모량을 추정하기 위하여 사용할 실행 속도를 결정하는 제2 단계;

각각의 실시간 작업(T^m)에 대하여 주어진 데드라인(D^m)에 대해 상기 결정된 실행 속도들의 변환 시점을 결정하는 제3 단계; 및

상기 결정된 변환 시점에 기초하여 각각의 실시간 작업(T^m)들의 에너지 소모량의 합이 최소값을 가지도록

각 실시간 작업(T^m)의 할당 시간(A^m)을 결정하는 제4 단계

를 포함하는 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법.

청구항 2

제1항에 있어서,

상기 제1 단계에서, 상기 확률적 계산량 모델(Φ_c^m)은, m번째 작업(T^m)에 대하여

$$\Phi_c^m = (1 - \sum_{i=1}^c p_i^m)$$
 (여기서, P_i^m 은 m번째 작업이 c번째 주기에서의 누적 확률임)의 수식에 의하여 결정되는 것을 특징으로 하는 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법.

청구항 3

제1항에 있어서,

상기 제2 단계는, 사용가능한 K개의 이산 주파수(discrete frequency) 중에서

$$\frac{P_y - P_x}{F_y - F_x} > \frac{P_z - P_y}{F_z - F_y} \quad \text{for } F_x < F_y < F_z$$
 (여기서, F_x, F_y, F_z 는 사용가능한 이산 주파수이고

P_x, P_y, P_z 는 각 주파수에 대응하는 에너지 소모량)의 수식을 만족하는 주파수(F_y)를 비효율적인 실행속도를 갖는 주파수로 결정하고 이들을 제거하는 것을 특징으로 하는 변동하는 계산량을 갖는 실시간 작업을 위한

단일 코어 프로세서의 에너지 효율적 스케줄링 방법.

청구항 4

제1항에 있어서,

상기 제3 단계에서 상기 전환 시점(Φ_{π_k})들은

$$\Phi_{\pi_k} = \Phi_{\pi_K} \cdot \frac{P_K - P_{K-1}}{1/F_{K-1} - 1/F_K} \cdot \frac{1/F_{k-1} - 1/F_k}{P_k - P_{k-1}}$$

for $K > k > 1$ (여기서, $\Phi_{\pi_k}, \Phi_{\pi_K}$ 는 각각 π_k, π_K 에서의 확률적 계산량 모델이고,

$P_k, P_{k-1}, P_K, P_{K-1}$ 은 각각 $k, k-1, K, K-1$ 에서의 에너지 소모량이고,

$F_{k-1}, F_k, F_{K-1}, F_K$ 는 각각 $k-1, k, K, K-1$ 에서의 주파수임)의 수식에 의해 계산되는 것을 특징으로 하는 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법.

청구항 5

제1항에 있어서,

상기 제4 단계는,

각 실시간 작업(T^m)의 할당 시간(A^m)을 결정하기 위하여, 각 작업들에 대해 가장 빠른 실행 속도의 주파수에 대응하는 작업 시간을 할당하는 단계;

나머지 사용가능한 시간을 균등하게 할당할 때의 각 작업의 에너지 감소율을 계산하는 단계; 및

최대 에너지 감소율을 갖는 작업을 결정하고 이에 대해서만 추가 시간을 할당하는 단계

를 모든 작업에 대해 더 이상 할당할 시간이 없을 때까지 반복함으로써 각 작업의 할당 시간(A^m)을 결정하는 것을 특징으로 하는 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법.

명세서

기술분야

[0001] 본 발명은 프로세서의 스케줄링 방법에 관한 것으로서 보다 상세하게는 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법에 관한 것이다.

배경기술

[0002] 동적 전압 및 주파수 스케일링(Dynamic voltage and frequency scaling, DVFS) 기법에서는 프로세서의 코어에

적용되는 전압값을 증가시키면 클럭 주파수가 정비례하여 증가하고 또한 프로세서의 코어의 실행 속도 또한 증가한다. 프로세서 코어에 적용되는 전압(클럭주파수 값 또는 코어의 실행속도)를 증가시키면 에너지 소모량도 이에 비례하여 증가하고 일반적으로 에너지 소모량 증가율이 전압값 증가율보다 크다.

[0003] 이러한 관계를 이용하여 실시간 작업(real-time tasks)의 데드라인(deadline)에서 최대 계산량(worst-case computation amount)을 종료하도록 클럭 주파수를 감소시키면서 에너지 소모량을 최소화하기 위한 다양한 방법이 제안되어 있다.

[0004] 고정된 최대 계산량이 아닌 변동하는 계산량을 고려하는 방안이 최근 제안되어 있다. 실제 계산량은 대부분의 경우 최대 계산량보다 작으며 또한 종료시까지 불확실한 경우가 대부분이다. 이러한 방안들은 변동하는 계산량을 확률적 계산량으로 변환하여 확률적 계산량의 평균 에너지 소모량을 최소화한다. 낮은 에너지 소모량을 갖는 낮은 클럭 주파수가 높은 확률을 갖는 연산부에 할당되고 높은 클럭 주파수는 낮은 확률을 갖는 연산부에 할당된다. 실제 계산량이 최대 계산량보다 작은 경우 이러한 확률적 계산 기법은 데드라인에 정확히 최대 계산량을 종료하는 고정 주파수를 할당하는 종래의 방식에 비해 보다 적은 에너지를 소모한다. 이러한 기법들은 클럭 주파수와 에너지 소모량 사이의 수식에 의해 무한 연속 주파수를 갖는 실시간 작업들의 에너지 소모를 최소화하는 문제를 해결하고 있다.

[0005] 그러나, 실제의 DVFS 기법의 시스템 환경에서는 유한한 클럭 이산 주파수(discrete frequency)가 제공되며 적용 가능한 클럭 이산 주파수와 에너지 소모량과의 관계 또한 불규칙하다. 또한, 종래의 기술은 단일 실시간 작업(single real-time task)의 경우만을 고려하고 있다.

[0006] 본 출원의 발명자가 제안했던 대한민국 공개특허공보 제10-2010-0026989호(2010.03.10)는 "에너지 절약형 멀티코어 프로세서"에 대하여 개시하고 있으나 이는 고정된 계산량(worst-case 최대 계산량)을 기준으로 데드라인 이내에 수행을 완료하는 에너지 소모량을 최소화하는 방법을 개시하고 있다는 점에서 한계점을 가지고 있다.

[0007] 또한, 본 출원의 발명자가 또 다시 제안했던 대한민국 등록특허공보 제10-1065436호(2011.09.19. 공개)는 "불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법"에 관한 것으로서, 이는 멀티코어 프로세서상에서 병렬로 수행가능한 단일 실시간 작업의 변동하는 불확실한 계산량을 확률적 계산량 모델로 표현하여 에너지 소모량의 확률적 기대값을 최소화하는 방법에 대해 개시하고 있으나, 단일코어 프로세서상에서 변동하는 불확실한 계산량을 갖는 다수의 실시간 작업들에 대해서는 적용할 수 없다는 한계점을 갖는다.

선행기술문헌

특허문헌

- [0008] (특허문헌 0001) 대한민국 공개특허공보 제10-2010-0026989호(2010.03.10. 공개)
(특허문헌 0002) 대한민국 등록특허공보 제10-1065436호(2011.09.19. 공고)

발명의 내용

해결하려는 과제

[0009] 본 발명은 상기한 바와 같은 한계점을 감안하여 안출된 것으로서, 단일코어 프로세서상에서 변동하는 불확실한 계산량을 갖는 다수의 실시간 작업들에 대해서 불확실한 계산량을 확률적 계산량 모델로 표현하고 확률적 계산량 모델에 기초하여 다수의 실시간 작업들의 에너지 소모량의 합을 최소화하는 스케줄링 방법을 제공하는 것을 목적으로 한다.

과제의 해결 수단

[0010] 상기한 바와 같은 과제를 해결하기 위하여 본 발명은, 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어

프로세서의 에너지 효율적 스케줄링 방법에 있어서, M개의 실시간 작업(T^m) 각각에 대하여 각 작업의 데드라인(D^m) 이전에 수행해야 할 계산량을 예측하기 위한 확률적 계산량 모델(Φ_c^m)을 결정하는 제1 단계; 각각의 실시간 작업(T^m)에 대해 상기 확률적 계산량 모델(Φ_c^m)에 의한 에너지 소모량을 추정하기 위하여 사용할 실행 속도를 결정하는 제2 단계; 각각의 실시간 작업(T^m)에 대하여 주어진 데드라인(D^m)에 대해 상기 결정된 실행 속도들의 변환 시점을 결정하는 제3 단계; 및 상기 결정된 변환 시점에 기초하여 각각의 실시간 작업(T^m)들의 에너지 소모량의 합이 최소값을 가지도록 각 실시간 작업(T^m)의 할당 시간(A^m)을 결정하는 제4 단계를 포함하는 변동하는 계산량을 갖는 실시간 작업을 위한 단일 코어 프로세서의 에너지 효율적 스케줄링 방법을 제공한다.

[0011] 여기에서, 상기 제1 단계에서, 상기 확률적 계산량 모델(Φ_c^m)은, m번째 작업(T^m)에 대하여

$$\Phi_c^m = (1 - \sum_{i=1}^c p_i^m) \quad (여기서, P_i^m \text{은 } m\text{번째 작업이 } c\text{번째 주기에서의 누적 확률임})$$

의 수식에 의하여 결정될 수 있다.

[0012] 또한, 상기 제2 단계는, 사용가능한 K개의 이산 주파수(discrete frequency) 중에서

$$\frac{P_y - P_x}{F_y - F_x} > \frac{P_z - P_y}{F_z - F_y} \quad \text{for } F_x < F_y < F_z$$

(여기서, F_x, F_y, F_z 는 사용가능한 이산 주파수이고 P_x, P_y, P_z 는 각 주파수에 대응하는 에너지 소모량)의 수식을 만족하는 주파수(F_y)를 비효율적인 실행속도를 갖는 주파수로 결정하고 이들을 제거하도록 구성할 수 있다.

[0013] 또한, 상기 제3 단계에서 상기 전환 시점(Φ_{π_k})들은

$$\Phi_{\pi_k} = \Phi_{\pi_{K-1}} \cdot \frac{P_K - P_{K-1}}{1/F_{K-1} - 1/F_K} \cdot \frac{1/F_{K-1} - 1/F_K}{P_K - P_{K-1}}$$

[0014]

for $K > k > 1$ (여기서, $\Phi_{\pi_k}, \Phi_{\pi_K}$ 는 각각 π_k, π_K 에서의 확률적 계산량 모델이고, $P_k, P_{k-1}, P_K, P_{K-1}$ 은 각각 $k, k-1, K, K-1$ 에서의 에너지 소모량이고, $F_{k-1}, F_k, F_{K-1}, F_K$ 는 각각 $k-1, k, K, K-1$ 에서의 주파수임)의 수식에 의해 계산될 수 있다.

[0016] 또한, 상기 제4 단계는, 각 실시간 작업(T^m)의 할당 시간(A^m)을 결정하기 위하여, 각 작업들에 대해 가장 빠른 실행 속도의 주파수에 대응하는 작업 시간을 할당하는 단계; 나머지 사용가능한 시간을 균등하게 할당할 때의 각 작업의 에너지 감소율을 계산하는 단계; 및 최대 에너지 감소율을 갖는 작업을 결정하고 이에 대해서만 추가 시간을 할당하는 단계를 모든 작업에 대해 더 이상 할당할 시간이 없을 때까지 반복함으로써 각 작업

의 할당 시간(A^m)을 결정하도록 구성할 수도 있다.

발명의 효과

- [0017] 본 발명에 의하면, 단일코어 프로세서상에서 변동하는 불확실한 계산량을 갖는 다수의 실시간 작업들에 대해서 불확실한 계산량을 확률적 계산량 모델로 표현하고 확률적 계산량 모델에 기초하여 다수의 실시간 작업들의 에너지 소모량의 합을 최소화하는 스케줄링 방법을 제공할 수 있다.

도면의 간단한 설명

- [0018] 도 1은 본 발명에 의한 변동하는 계산량을 갖는 실시간 작업(real-time Task)을 위한 단일 코어 프로세서(single core processor)의 에너지 효율적 스케줄링 방법의 일실시예를 나타낸 흐름도이다.

도 2 및 도 3은 변동하는 계산량(computation amount)의 통계적 모델을 나타낸 도면으로서, 여기에서 계산량은 작업을 종료할 때 까지 필요한 프로세서의 클럭 주기(clock cycle)의 개수로 표현되어 있다. 도 2는 필요한 클럭 주기의 확률 분포(probability distribution)를 나타낸 것이고 도 3은 도 2의 확률 분포의 테일 누적 분포(tail culmulative distribution)를 나타낸 것이다.

도 4는 단계(S200)에 의해 제거되고 남은 주파수들과 에너지 소모량 사이의 관계를 나타낸 도면이다.

도 5는 각각의 실시간 작업(T^m)들의 에너지 소모량의 합이 최소값을 가지도록 각 실시간 작업(T^m)의 할당 시간(A^m)을 결정하는 과정 즉 단계(S400)을 나타낸 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0019] 이하 첨부 도면을 참조하여 본 발명에 의한 실시예를 상세하게 설명하도록 한다.

- [0020] 도 1은 본 발명에 의한 변동하는 계산량을 갖는 실시간 작업(real-time Task)을 위한 단일 코어 프로세서(single core processor)의 에너지 효율적 스케줄링 방법의 일실시예를 나타낸 흐름도이다.

- [0021] 도 1을 참조하면, 본 실시예에 의한 방법은, 우선 M개의 실시간 작업(T^m) 각각에 대하여 각 작업의 테드라인(D^m) 이전에 수행해야 할 계산량을 예측하기 위한 확률적 계산량 모델(Φ^m_c)을 결정한다(S100).

- [0022] 본 실시예에서, 실시간 작업(real-time Tasks)은 M개의 복수개로 구성되며, 이들 중에서 m번째의 작업을 T^m 으로 나타낸다. 각각의 작업(T^m)은 작업을 끝내야하는 자신의 테드라인(deadline)을 가지며 이를 D^m 으로 나타낸다. 각 작업(T^m)에 대해 종료시까지 소요되는 계산량은 정확히 알 수 없지만 즉, 불확실하지만 본 발명에서는 이를 확률적 계산량 모델에 의하여 온라인 또는 오프라인 프로파일링에 의해 추정하는 방법을 사용한다.

- [0023] 도 2 및 도 3은 변동하는 계산량(computation amount)의 통계적 모델을 나타낸 도면으로서, 여기에서 계산량은 작업을 종료할 때 까지 필요한 프로세서의 클럭 주기(clock cycle)의 개수로 표현되어 있다. 도 2는 필요한 클럭 주기의 확률 분포(probability distribution)를 나타낸 것이고 도 3은 도 2의 확률 분포의 테일 누적 분포(tail culmulative distribution)를 나타낸 것이다.

[0024] 도 2 및 도 3을 참조하면, c번째 주기(c^{th} cycle)에서의 확률은 P_c 로 나타낼 때, c번째 주기에서의 누적 확률(cumulative probability)은 $\sum_{i=1}^c p_i$ 이며 또한 c번째 주기에서의 테일 누적 확률은 $1 - \sum_{i=1}^c p_i$ 이다.

[0025] 여기서, $c_1 < c_2$ 에 대해서 $\Phi_{c_1}^m \geq \Phi_{c_2}^m$ 임을 유의해야 하는데, 이는 확률 함수의 누적 분포는 비감소(non-decreasing)이고 따라서 이들의 테일 분포는 비증가(non-increasing)이기 때문이다.

[0026] 작업(T^m)에 대한 요구되는 계산 주기의 최대값(worst-case number of required computation cycle)는 W^m 으로 나타낸다. 각각의 작업들은 서로 다른 D^m, W^m, Φ_c^m 을 갖는데 이들은 미리 스케줄러에게 알려져 있는 값이다.

[0027] 이상과 같은 방법에 의하여 단계(S100), 즉, M개의 실시간 작업(T^m) 각각에 대하여 각 작업의 테드라인(D^m) 이전에 수행해야 할 계산량을 예측하기 위한 확률적 계산량 모델(Φ_c^m)을 결정한다.

[0028] 본 발명에서 단일 코어 프로세서는 사용가능한 실행 속도 즉, 주파수(frequency)로서 K개의 이산 주파수(discrete frequency)를 갖는 것을 전제로 하며 이들은 각각 증가하는 순서대로 $F_1, F_2, \dots, F_K$ 로 나타낸다. 이들의 에너지 소모량(power consumption)은 P_i 로 나타낸다. 여기서, $F_i < F_j$ 이면 $P_i < P_j$ 이다. F_i 와 P_i 는 특별한 관계를 갖지는 않는다. 실행속도(F_i)에서의 각 주기(cycle)의 실행 시간(execution time)은 $\frac{1}{F_i}$ 이다. 클럭 주파수의 전환의 오버헤드(overhead)는 무시할 수 있는 것으로 가정한다.

[0029] 작업(T^m)의 c번째 주기에 할당되는 실행속도 즉, 주파수는 f_c^m 으로 나타내는데 여기서 $f_c^m \in \{F_1, \dots, F_K\}$ 이다. 이들의 에너지 소모량(power consumption)은 $P(f_c^m)$ 으로 나타내는데 여기서 $P(F_1) = P_1, \dots, P(F_K) = P_K$ 이다.

[0030] 이러한 상태에서 주어진 데드라인 조건을 충족하면서 단일 작업(T^m)의 평균 에너지 소모량을 최소화하는 것은 다음과 같은 식으로 표현될 수 있다.

[0031] [수학식 1]

$$\sum_{c=1}^{W^m} P(f_c^m) \cdot \Phi_c^m$$

[0032] Minimize

$$\frac{\sum_{c=1}^{W^m} \frac{1}{f_c^m}}{D^m} \leq 1$$

[0033] subject to

[0034] M개의 작업들에 대해 상기 수학식 1의 문제를 확장하면 다음과 같이 표현할 수 있다.

[0035] [수학식 2]

$$\sum_{m=1}^M \sum_{c=1}^{W^m} P(f_c^m) \cdot \Phi_c^m$$

[0036] Minimize

$$\sum_{m=1}^M \frac{\sum_{c=1}^{W^m} \frac{1}{f_c^m}}{D^m} \leq 1$$

[0037] subject to

[0038] 이러한 수학식에 대해 유도되는 해(solution)은 f_c^m 을 각각의 m 및 $1 \leq c \leq W^m$ 에 대하여 결정하게 된다.

[0039] 이를 위하여 각각의 실시간 작업(T^m)에 대해 상기 확률적 계산량 모델(Φ^m)에 의한 에너지 소모량을 추정하기 위하여 사용할 실행 속도 즉, 주파수(frequency)를 결정한다(S200).

[0040] 사용가능한 주파수는 전술한 바와 같이 K개의 이산 주파수(discrete frequency)이며 단계(S200)에서는 이중에서 적절한 주파수를 선택하게 되는데 이는 비효율적인 실행속도(power-inefficient frequency)를 제거하는 것에 이

$$\frac{P_y - P_x}{F_y - F_x} > \frac{P_z - P_y}{F_z - F_y} \quad \text{for } F_x < F_y < F_z$$

루어질 수 있다. 비효율적인 실행 속도는 F_y 의 관계를

만족하는 F_y 로 정의할 수 있는데, 이러한 F_y 를 주어진 실행 속도로부터 제거한다. 상기 수식은 에너지 소모량의 볼록 함수(convex function)에 부합하지 않는 주파수를 의미한다.

$$\frac{P_k - P_{k-1}}{F_k - F_{k-1}} > \frac{P_{k+1} - P_k}{F_{k+1} - F_k} \quad \text{for } 1 < k < K$$

를 계산하면 모든 비효율적 실행 속도(주파수)를 선택할 수 있다. 이렇게 하면 에너지 소모량은 주파수의 볼록 함수 즉,

$$\frac{P_k - P_{k-1}}{F_k - F_{k-1}} \leq \frac{P_{k+1} - P_k}{F_{k+1} - F_k} \quad \text{for } 1 < k < K$$

가 된다.

[0041] 도 4는 단계(S200)에 의해 제거되고 남은 주파수들과 에너지 소모량 사이의 관계를 나타낸 도면이다. 도 4를 참조하면, 단위 계산량을 수행하는데 걸리는 시간이 커질수록 에너지 소모량이 적으며 단위 계산량을 수행하는데 걸리는 시간이 작을수록 에너지 소모량이 크다는 것을 나타내고 있다.

[0042] 전술한 바와 같은 과정을 통해 실행 속도 즉, 주파수들이 결정되면, 각각의 실시간 작업(T^m)에 대하여 주어진 테드라인(D^m)에 대해 상기 결정된 실행 속도들의 변환 시점(transition points)을 결정한다(S300).

[0043] 이는 $f_{c_1} \leq f_{c_2}$ for $1 \leq c_1 < c_2 \leq W$ 의 관계에 의하여 이루어질 수 있는데, 이러한 정의에 의하여 낮은 주파수에서 높은 주파수로의 전환 시점(transition point)만을 고려하는 것으로 충분하게 된다.

[0044] 우선, F_{k-1} 에서 F_k 로의 전환 시점(F_k 의 시작 주기(starting cycle))을 π_k 로 나타낸다. $f_{c_1} \leq f_{c_2}$ for $1 \leq c_1 < c_2 \leq W$ 에 의하여 $\pi_k \leq \pi_{k+1}$ 을 의미한다. F_{k-1} 이 F_k 대신 F_{k+1} 로 전환되면 $\pi_{k+1} = \pi_k$ 가 된다. $F_k, \dots, F_K$ 가 사용되지 않는 경우에는 $\pi_k = \dots = \pi_K = (W+1)$ 이 된다. 그러면 수학식 1은 다음과 같은 수학식 3으로 표현될 수 있다.

[0045] [수학식 3]

[0046] Minimize

$$P_1 \cdot \sum_{c=\pi_1}^{(\pi_2-1)} \Phi_c + P_2 \cdot \sum_{c=\pi_2}^{(\pi_3-1)} \Phi_c + \dots + P_K \cdot \sum_{c=\pi_K}^W \Phi_c$$

[0047] subject to

$$\frac{\pi_2 - \pi_1}{F_1} + \frac{\pi_3 - \pi_2}{F_2} + \dots + \frac{(W+1) - \pi_K}{F_K} \leq D$$

[0048] 여기에서 F_1 은 최소 주파수이기 때문에 $\pi_1=1$ 이다.

[0049] 모든 π_k 들을 안다면 수학식 1의 해를 직접 구할 수 있으나 π_k 들의 값은 주어진 테드라인(D)에 의존한다. π_k 를 구하기 위하여 우선 π_k 들의 값 사이의 관계를 알아보고 이를 이용하여 테드라인(D)와 관련된 π_k 를 구하는데 사용한다.

[0050] 다음의 정리(theorem)은 π_k 들 사이의 관계를 나타낸다.

[0051] [정리 1]

$$\Phi_{\pi_x} \cdot \frac{P_x - P_{x-1}}{1/F_{x-1} - 1/F_x} = \Phi_{\pi_y} \cdot \frac{P_y - P_{y-1}}{1/F_{y-1} - 1/F_y} \quad \text{for } 2 \leq x < y \leq K$$

[0052]

[0053] 고정된 π_K 로부터 상기 정리 1의 관계를 만족시키는 $\pi_K, \dots, \pi_2$ 를 구할 수 있다. $\pi_K = 1$ 부터

$\pi_K = W$ 까지에 대해 $\pi_{K-1}, \dots, \pi_2$ 를 반복적으로 계산함으로써 테드라인(D)에 매칭되는 정확한 π_k 들을 구할 수 있게 된다.

[0054] π_k 값들을 찾는 과정은 다음과 같다.

[0055] 우선, $\pi_K = 1$ 을 할당하고,

$$\Phi_{\pi_k} = \Phi_{\pi_K} \cdot \frac{P_K - P_{K-1}}{1/F_{K-1} - 1/F_K} \cdot \frac{1/F_{k-1} - 1/F_k}{P_k - P_{k-1}}$$

[0056]

[0057] for $K > k > 1$

[0058] 의 관계를 만족하는 전환 시점(π_k)를 찾는다.

[0059] Φ_c 의 정의에 의해 $0 \leq \Phi_{\pi_k} \leq 1$ 이기 때문에 $P_{\pi_k} > 1$ 인 π_k 는 1로 설정된다.

[0060] 이는 검색되는 스케줄의 실행 시간을 계산하고 이를 주어진 테드라인(D)와 비교하게 된다. 검색된 스케줄의 실행 시간이 테드라인보다 작은 경우에는 π_K 의 값을 증가시키고 다시 $K > k > 1$ 에 대해 전환 시점(π_k)를 검색한다.

[0061] $\pi_K = W$ 이지만 실행시간이 그래도 테드라인보다 작은 경우에는 π_K 를 $(W+1)$ 로 고정된 후 π_{K-1}

의 값을 증가시키고 $(K-1) > k > 1$ 에 대해서 다시 나머지 전환 시점 π_k 를 검색한다. 마찬가지로,

$\pi_j = W$ 이지만 실행 시간이 아직 테드라인보다 작은 경우에는 π_j 의 값을 $(W+1)$ 로 고정된 후 π_j

대신 π_{j-1} 의 값을 증가시킨다. 이러한 과정을 검색되는 스케줄의 실행이 테드라인과 같아질때까지 반복한다.

[0062] 이러한 과정을 수행하게 되면 각각의 작업(T^m)에 대해 수학식 1, 즉 수학식 3을 만족시키는 값, 즉, 각 작

업(T^m)에 대해 예상되는 에너지 소모량을 최소화할 수 있는 확률적 기대값을 계산할 수 있게 된다.

[0063] 다음으로, 상기 결정된 변환 시점에 기초하여 각각의 실시간 작업(T^m)들의 에너지 소모량의 합이 최소값을 가지도록 각 실시간 작업(T^m)의 할당 시간(A^m)을 결정한다(S400).

[0064] 전술한 바와 같이 단계(S300)까지를 수행하게 되면, 주어진 Φ_c^m 및 D^m 에 대하여 각 작업(T^m)에 대해 예상되는 에너지 소모량을 최소화할 수 있는 확률적 기대값을 계산할 수 있는데 이는 각각의 작업(T^m)에 대해 주어진 할당 시간(allocated time, A^m)에 의존하는 값이다. 여기에서 $A^m = \delta^m \cdot D^m$ 이며 $0 < \delta^m \leq 1$ 및 $\sum_{m=1}^M \delta^m \leq 1$ 이다.

[0065] 주어진 할당 시간(A^m) 즉, $\delta^m \cdot D^m$ 에 대해 전술한 수학식 3의 최소값을 얻으면 수학식 2의 결과는 전적으로 δ^m 들의 결정에 의하여 이루어진다.

[0066] 결정된 δ^m 에 의해 수학식 3에서 유도된 최고값을 Ψ^m 으로 나타내면, 수학식 2는 다음과 같이 표현할 수 있다.

[0067] [수학식 4]

[0068] Minimize $\sum_{m=1}^M \Psi^m$ subject to $\sum_{m=1}^M \delta^m \leq 1$

[0069] 할당 시간($A^m = \delta^m \cdot D^m$)이 증가함에 따라 에너지 소모량(Ψ^m)은 감소하고 각 작업(T^m)에 대해 단위 시간 당 에너지 소모량의 감소비(decrement ratio of power consumption per unit time)($\frac{\Psi^m}{\partial A^m}$)은 감소한다. 이 경우 상기 수학식 4의 최소값은, 에너지 소모량의 최대 감소를 발생시키는 추가 단위 시간을 작업에 증가적으로(incrementally) 할당함으로써 이루어질 수 있다. 수학식 4의 최소값은

$$\sum_{m=1}^M \frac{A^m}{D^m} = \sum_{m=1}^M \delta^m = 1$$

일 때 발생한다.

[0070] 도 5는 각각의 실시간 작업(T^m)들의 에너지 소모량의 합이 최소값을 가지도록 각 실시간 작업(T^m)의 할당 시간(A^m)을 결정하는 과정 즉 단계(S400)을 나타낸 흐름도로서, 이하 도 5를 참조하여 단계(S400)을 설명한다.

[0071] 단계(S400)은 M개의 작업에 사용 가능한 시간을 할당하여 할당된 시간에 의해 획득되는 에너지 소모량의 전체 감소율을 최대화하는 과정으로서, 우선, 모든 계산에 대해 적어도 가장 빠른 주파수를 제공할 수 있도록

$\frac{W^m}{F_K}$ 가 각각의 A^m 에 할당된다(S410).

[0072] 다음으로, 모든 작업에 대해 나머지 사용가능한 시간을 균등하게 할당할 때 각 작업의 에너지 감소율을 계산한다(S420).

[0073] 즉, x 가 각 작업에 대해 동시에 또한 추가적으로 할당가능한 최대 시간이라고 하면,

$$\frac{A^{1+x}}{D^1} + \dots + \frac{A^{M+x}}{D^M} = 1 \quad x = \frac{1 - \sum_{m=1}^M A^m / D^m}{\sum_{m=1}^M 1 / D^m}$$

이며 따라서 이다.

[0074] 상기 식은 각 작업(T^m)에 대해 시간 A^m 을 할당할 때와 시간 $(A^m + x)$ 를 할당할 때의 에너지 소모량 사이의 감소율을 계산한다. 그리고 최대 에너지 감소율을 갖는 작업을 선택하고 이 작업에 대해서만 추가 시간 x 를 할당한다(즉, $A^m \leftarrow (A^m + x)$)(S430).

[0075] 이와 같은 과정을 모든 작업에 대해 더 이상 할당할 시간이 없을 때까지(즉, $\sum_{m=1}^M \frac{A^m}{D^m} = \sum_{m=1}^M \delta^m = 1$) 반복한다.

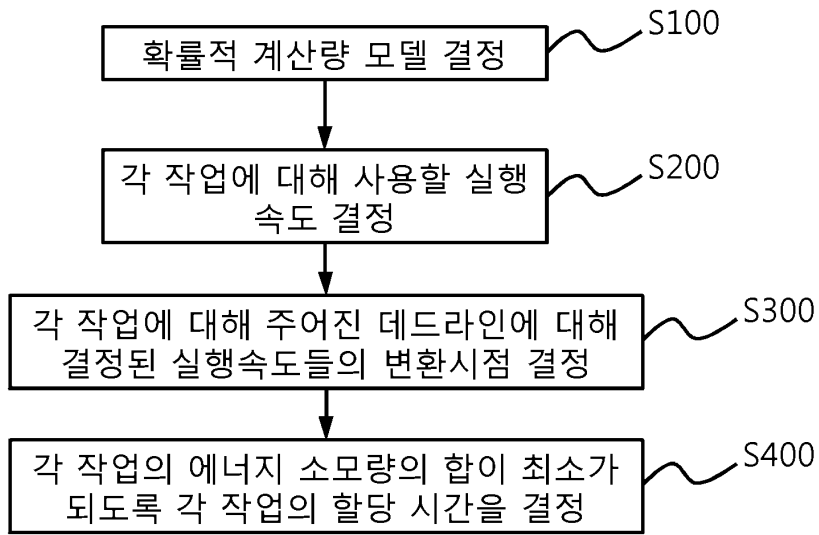
[0076] 이와 같은 과정을 수행함으로써 δ^m 값을 점진적으로 증가시키면서 $\sum \Psi^m$ 을 최소화, 즉 상기 수학식 4를 최소화할 수 있게 된다. 즉, 각각의 작업(T^m)에 대해 결정된 주파수들의 전환 시점에 따라 M개의 작업들의 확률적 에너지 소모량의 합을 최소화할 수 있도록 각각의 작업들의 할당 시간을 결정할 수 있게 되므로 결국 M개의 작업들의 전체 에너지 소모량의 합을 최소화할 수 있는 스케줄링 구조를 생성하게 된다.

[0077] 이러한 스케줄링 구조에 의해 각각의 작업(T^m)에 대해 주파수(f_c^m)을 결정한다. 프로세서의 스케줄러는 복수개의 작업 중에서 가장 빠른 데드라인을 갖는 작업(T^e)을 우선적으로 실행하고 작업(T^e)의 종료시까지 결정된 주파수(f_c^e)를 할당하게 된다.

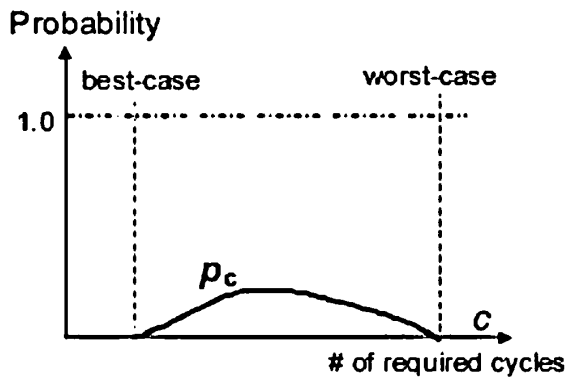
[0078] 이상에서, 바람직한 실시예들을 참조하여 본 발명을 설명하였으나, 본 발명은 상기의 실시예들에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능할 것이다. 따라서 본 발명은 첨부한 특허청구범위 및 도면 등의 전체적인 기재를 참조하여 해석되어야 할 것이며, 이의 균등 또는 등가적 변형 모두는 본 발명 사상의 범주에 속한다고 할 것이다.

도면

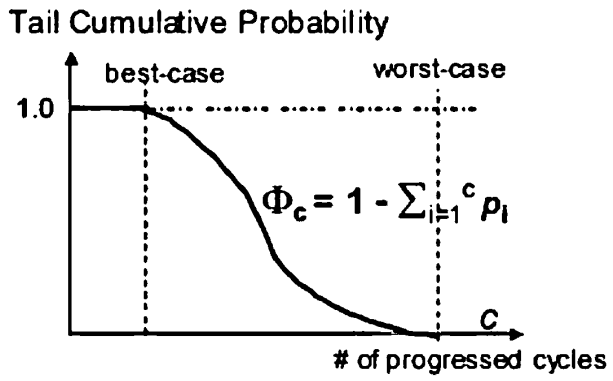
도면1



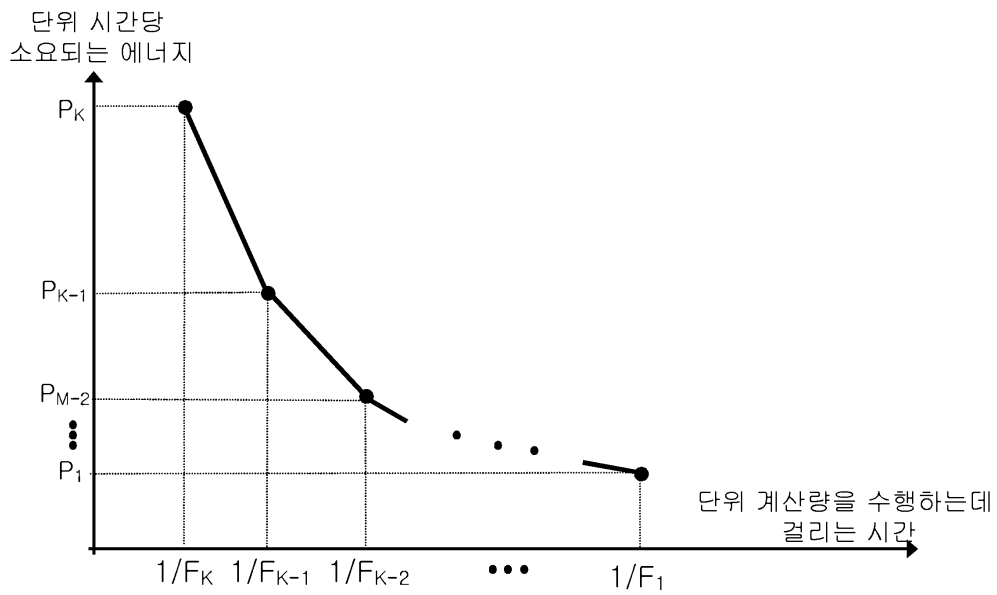
도면2



도면3



도면4



도면5

