



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년09월19일
(11) 등록번호 10-1065436
(24) 등록일자 2011년09월08일

(51) Int. Cl.

G06F 9/50 (2006.01) G06F 9/28 (2006.01)

(21) 출원번호 10-2010-0124251

(22) 출원일자 2010년12월07일

심사청구일자 2010년12월07일

(56) 선행기술조사문헌

KR101034409 B1

KR100982333 B1

KR1020060080666 A

대한전자공학회 2009년 SoC학술대회, 정일중 외3
인 공저 휴대장치를 위한 확률적 태스크 스케줄링
방법론, 페이지526-529 (2009.05.)

전체 청구항 수 : 총 7 항

(73) 특허권자

경상대학교산학협력단

경상남도 진주시 가좌동 900

(72) 발명자

이완연

강원도 춘천시 칠전동 대우아파트 205동 602호

김경훈

경상남도 진주시 상대2동 301-94번지 202호

(74) 대리인

조흠오

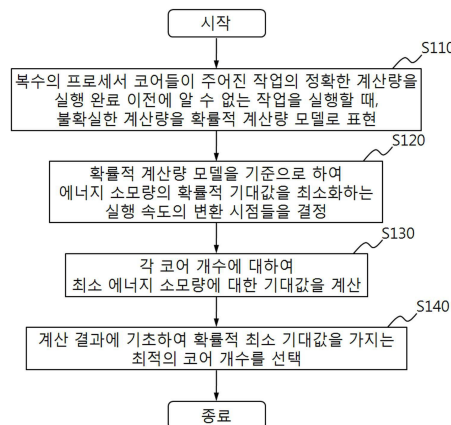
심사관 : 황승희

(54) 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법

(57) 요약

본 발명은 확률적 스케줄링 방법 및 이를 이용하는 저전력 멀티 코어 프로세서를 제공한다. 확률적 스케줄링 방법은 복수의 프로세서 코어들이 주어진 작업의 정확한 계산량을 실행 완료 이전에 알 수 없는 작업을 실행할 때, 불확실한 계산량을 확률적 계산량 모델로 표현하고 이 확률적 계산량 모델을 기준으로 하여 에너지 소모량의 확률적 기대값을 최소화하는 실행 속도의 변환 시점들을 결정하고, 각 코어 개수에 대하여 최소 에너지 소모량에 대한 기대값을 계산하고, 계산 결과에 기초하여 확률적 최소 기대값을 가지는 최적의 코어 개수를 선택하는 것을 특징으로 한다.

대표도 - 도1



특허청구의 범위

청구항 1

복수의 프로세서 코어들이 주어진 작업을 데드라인(deadline) 이전에 완료하는데 소요되는 정확한 계산량을 실행 완료 이전에 알 수 없는 불확실한 계산량을 갖는 작업을 실행할 때, 상기 주어진 작업을 데드라인 이전에 완료하는데 소요되는 계산량을 상기 주어진 작업의 계산량의 확률적 분포 함수에 대한 누적 함수의 차함수인 확률적 계산량 모델로 표현하는 제1 단계;

상기 확률적 계산량 모델을 기준으로 하여 에너지 소모량의 확률적 기대값을 최소화하는 실행 속도의 변환 시점들을 결정하는 제2 단계;

상기 복수의 프로세서 코어들의 각 코어 개수에 대하여 최소 에너지 소모량에 대한 기대값을 계산하는 제3 단계; 및

상기 제3 단계의 계산 결과에 기초하여 확률적 최소 기대값을 가지는 최적의 코어 개수를 선택하는 제4 단계를 포함하는 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 제3 단계는 상기 작업을 수행하는 코어의 개수와 계산 진행량에 따른 실행 속도를 결정하기 위하여, 고정된 n 의 값에 대하여 아래의 수학식 3에 따라 해법을 찾는 단계를 포함하고,

[수학식 3]

$$e_1 \cdot \int_0^{c[1]} P_t^n \cdot dt + e_2 \cdot \int_0^{c[2]} P_t^n \cdot dt + \dots + e_M \cdot \int_{c[M-1]}^{c[M]} P_t^n \cdot dt$$

여기에서, 수학식 3은 아래의 수학식 4의 조건을 구비하며,

[수학식 4]

$$1/f_1 \cdot (c[1] - 0) + 1/f_2 \cdot (c[2] - c[1]) + \dots + 1/f_M \cdot (c[M] - c[M-1]) \leq D$$

여기에서, 실행 속도 f_m 을 적용하다가 다음 실행 속도 $f_{(m+1)}$ 를 적용하는 시점을 $c[m]$ 이라 하고, $c[m]$ 은 $c[m+1]$ 과 같거나 작고, 데드라인에 수행을 완료하면서 에너지 소모량의 확률적 기대값을 최소화하도록 상기 $c[m]$ 값들이 결정되고,

N 개의 코어들 가진 프로세서에 있어서, n 개의 코어를 사용할 때, 여기에서 n 은 1 이상 N 이하의 자연수이고, $N-n$ 개의 나머지 코어는 전원을 끌고, 계산량 확률 분포는 P^n , 최대(worst-case) 계산량은 W^n , 그리고 데드라인은 D 로 각각 표현되고,

M 개의 적용 가능한 실행 속도는 각각 $f_1, f_2, \dots, f_M$ 으로 표현되고, 여기에서 M 은 2보다 큰 자연수이고, 각 실행 속도의 관계는 $f_1 < f_2 < \dots < f_M$ 을 만족하고,

f_m 의 실행속도에서 각 단계의 계산(a single cycle)을 수행하는데 필요한 에너지 소모량은 e_m 으로 표현되며, 여기서, m 은 1 이상, M 이하의 자연수이며, 각 실행 속도에 대한 에너지 소모량의 관계는 $e_1 < e_2 < \dots < e_M$ 을

만족하며,

상기 수학적 식 3에 의해 N개의 해법이 찾아지면, 상기 제4 단계는 상기 N개의 해법 중에서 최소 에너지 소모량의 기대값을 가지는 최적의 코어 개수를 선택하는 것을 특징으로 하는 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법.

청구항 4

삭제

청구항 5

제3항에 있어서,

상기 제2 단계는, 아래의 수학적 식 5의 관계를 가진 특정 실행 속도를 제거하는 단계를 포함하는 것을 특징으로 하는 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법:

[수학적 식 5]

$$\frac{(e_j - e_i)}{(1/f_i - 1/f_j)} > \frac{(e_k - e_j)}{(1/f_j - 1/f_k)}$$

여기에서, f_j 는 상기 특정 실행 속도를 나타내고, $f_i < f_j < f_k$ 의 관계를 만족하며, i , j , 및 k 는 자연수이다.

청구항 6

제5항에 있어서,

상기 제2 단계는, 상기 수학적 식 5의 관계를 가진 특정 실행 속도를 제거하는 단계 후에 남은 M개의 실행속도들의 변환 시점을 아래의 수학적 식 8에 기초하여 결정하는 단계를 포함하는 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법:

[수학적 식 8]

$$P_{c[y]} = P_{c[x]} \cdot \frac{[e_{(x+1)} - e_x]}{[1/f_x - 1/f_{(x+1)}]} \cdot \frac{[1/f_y - 1/f_{(y+1)}]}{[e_{(y+1)} - e_y]}$$

여기에서, $c[m]$ 또는 $c[y]$ 는 상기 변환 시점을 나타내고, m 또는 y 는 1 이상, M 미만의 자연수이다.

청구항 7

제6항에 있어서,

상기 제2 단계는 기준값 $c[1]$ 을 1부터 W^n 까지 차례로 고정하면서 상기 수학적 식 8을 만족하는 $c[2]$, $c[3]$, ..., $c[M]$ 의 값들을 찾고,

상기 기준값 $c[1]=1$ 로 찾아진 $c[m]$ 값들을 사용한 수행 완료시간을 아래의 수학적 식 9를 이용하여 계산하고,

[수학식 9]

$$\sum_{1 \leq m \leq M} \frac{1}{f_2} \cdot (c[m-1] - c[m])$$

상기 수학식 9의 계산 결과, 수행완료시간이 데드라인보다 크면 기준값을 $c[2]$ 로 교체한 후 $C[2]$ 를 1부터 W^n 까지 차례로 고정하면서 상기 수학식 8을 만족하는 $c[3]$, $c[4]$, ..., $c[M]$ 의 값들을 찾는 단계들을 포함하는 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법.

청구항 8

제7항에 있어서,

상기 제2 단계는 상기 수학식 9의 계산에 의해 찾아진 $c[m]$ 값들을 사용한 수행완료시간을 아래의 수학식 10을 이용하여 계산하고,

[수학식 10]

$$\sum_{b \leq m \leq M} \frac{1}{f_2} \cdot (c[m-1] - c[m])$$

상기 수학식 10에 의한 계산 결과를 데드라인(D) 값과 비교함으로써, 사용 가능한 $c[b]$ 값의 폭을 줄여가는 단계를 포함하는 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법.

청구항 9

제8항에 있어서,

상기 수학식 10의 이등분 검사 방식에 의해 찾아진 $c[b+1]$, ..., $c[M]$ 값들을 사용하여 아래의 수학식 11을 이용하여 예상 에너지 소모량 확률적 기대값을 계산하는 단계를 포함하는 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법:

[수학식 11]

$$n \cdot \sum_{b \leq m \leq M} e_m \cdot \int_{c[m-1]}^{c[m]} P_t^n \cdot dt$$

여기에서, $c[0]$ 은 0이다.

명세서

기술분야

[0001] 본 발명은 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법에 관한 것이다.

배경기술

[0002] 최근, 전기전자, 통신, 반도체 등의 기술 발전에 힘입어 컴퓨터, 이동통신 단말기, 스마트폰 등의 전자 기기가 폭넓게 보급되고 있다. 이러한 전자 기기는 사용자의 고성능 및 다기능 요구에 부응하기 위하여 멀티코어 프로세서를 구비할 수 있다.

[0003] 멀티코어 프로세서(Multi-Core Processor)는 2개 이상의 코어를 탑재하여 만든 프로세서이다. 멀티코어 프로세서를 이용하면, 단일 프로세서의 발열과 전력 문제를 어느 정도 해결할 수 있고, 복수의 작업을 각 코어에서 독립적으로 또는 병렬적으로 처리할 수 있다.

[0004] 더욱이, 서버 시스템이나 고성능 전자 기기의 진화와 함께 많은 장치 소프트웨어 또는 응용들(Applications)은 점차 높은 네트워크 대역폭 활용이나 멀티 태스크를 요구하고 있다. 이러한 요구를 만족하기 위하여 멀티코어를 효율적으로 활용하여 작업 또는 네트워크 처리율을 향상시키는 다양한 방안이 연구되고 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명의 목적은 멀티코어 프로세서상에서 불확실한 계산량을 가진 실시간 병렬 작업을 최소의 에너지 소모로 데드라인 이전에 수행을 완료할 수 있는 확률적 스케줄링 방법을 제공하는 것이다.

[0006] 또한, 본 발명의 목적은 예상 에너지 소모량을 확률적으로 최소화하도록 작업을 수행하는 코어 개수와 계산 진행량에 따른 실행 속도를 결정할 수 있는 확률적 스케줄링 방법을 제공하는 것이다.

과제의 해결 수단

[0007] 상기 기술적 과제를 해결하기 위하여 본 발명의 일 측면에 따르면, 복수의 프로세서 코어들이 주어진 작업의 정확한 계산량을 실행 완료 이전에 알 수 없는 작업을 실행할 때, 불확실한 계산량을 확률적 계산량 모델로 표현하는 제1 단계; 확률적 계산량 모델을 기준으로 하여 에너지 소모량의 확률적 기대값을 최소화하는 실행 속도의 변환 시점들을 결정하는 제2 단계; 복수의 프로세서 코어들의 각 코어 개수에 대하여 최소 에너지 소모량에 대한 기대값을 계산하는 제3 단계; 및 제3 단계의 계산 결과에 기초하여 확률적 최소 기대값을 가지는 최적의 코어 개수를 선택하는 제4 단계를 포함하는 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서의 확률적 스케줄링 방법이 제공된다.

발명의 효과

[0008] 본 발명에 의하면, 멀티코어 프로세서상에서 불확실한 계산량을 가진 실시간 병렬 작업을 최소의 에너지 소모로 데드라인 이전에 수행을 완료할 수 있다. 다시 말해서, 불확실한 계산량을 확률적 계산량 모델로 표현하고, 확률적 계산량 모델을 기준으로 하여 에너지 소모량의 확률적 기대값을 최소화하는 실행 속도와 코어 개수를 결정함으로써, 멀티코어 프로세서에서의 에너지 소모를 최소화할 수 있다.

[0009] 또한, 멀티코어 프로세서의 에너지 소모를 최소화할 수 있는 스케줄링 방법을 적용함으로써 전자 기기의 전력 소모를 최소화하고, 작업 수행의 안전성과 완전성을 높이고, 사용자의 편의성을 증대시킬 수 있다.

도면의 간단한 설명

[0010] 도 1은 본 발명의 일 실시예에 따른 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서에서의 확률적 스케줄링 방법(이하, 간단히 확률적 스케줄링 방법이라 한다)에 대한 개략적인 흐름도이다.

도 2a는 본 발명의 일 실시예에 채용가능한 일반적인 작업의 작업 종료시간의 확률적 분포를 나타낸 도면이다.

도 2b는 도 2a의 확률적 분포의 누적 함수를 나타낸 도면이다.

도 2c는 도 2b의 누적 함수의 차함수를 나타낸 도면이다.

도 3a 및 도 3b는 본 발명의 실시예에 따른 확률적 스케줄링 방법에 채용가능한 병렬처리 기반의 확률적 계산량 모델을 설명하기 위한 도면들이다.

도 4a 및 도 4b는 본 발명의 실시예에 따른 확률적 스케줄링 방법에 채용가능한 확률적 계산량 모델의 활용 방

안을 설명하기 위한 도면들이다.

도 5는 본 발명의 일 실시예에 따른 확률적 스케줄링 방법에서 비효율적 실행 속도를 제거하는 과정을 설명하기 위한 도면이다.

도 6a 및 도 6b는 본 발명의 일 실시예에 따른 확률적 스케줄링 방법에 채용가능한 실행속도의 변환시점을 찾는 방법을 설명하기 위한 도면들이다.

발명을 실시하기 위한 구체적인 내용

[0011] 이하, 첨부 도면을 참조하여 본 발명에 의한 실시예를 상세하게 설명하기로 한다.

[0012] 도 1은 본 발명의 일 실시예에 따른 불확실한 계산량을 가진 실시간 병렬 작업을 위한 멀티코어 프로세서에서의 확률적 스케줄링 방법(이하, 간단히 확률적 스케줄링 방법이라 한다)에 대한 개략적인 흐름도이다.

[0013] 도 1을 참조하면, 확률적 스케줄링 방법은, 복수의 프로세서 코어들이 주어진 작업의 정확한 계산량을 실행 완료 이전에 알 수 없는 작업을 실행할 때, 불확실한 계산량을 확률적 계산량 모델로 표현하는 제1 단계(S110), 확률적 계산량 모델을 기준으로 하여 에너지 소모량의 확률적 기대값을 최소화하는 실행 속도의 변환 시점들을 결정하는 제2 단계(S120), 복수의 프로세서 코어들의 각 코어 개수에 대하여 최소 에너지 소모량에 대한 기대값을 계산하는 제3 단계(S130), 및 제3 단계의 계산 결과에 기초하여 확률적 최소 기대값을 가지는 최적의 코어 개수를 선택하는 제4 단계를 구비한다. 각 단계는 이하의 설명에 의해 구체화될 것이다.

[0014] 확률적 계산량 모델은 일반적인 작업의 작업 종료시간의 확률적 분포를 누적 함수의 차함수로 나타냄으로써 표현될 수 있다. 여기에서, 확률적 계산량 모델은, 특정 작업이 특정 시간(t)에서 수행되고 있을 확률에 대한 함수를 지칭하는 것이다.

[0015] 기존의 방법은 최대 실행 시간(worst-case execution time)을 기반으로 에너지 소모량을 결정하였으나, 평균 실행 시간이 최대 실행 시간보다 현저히 짧은 경우, 사용 가능한 에너지 자원의 낭비가 크다. 따라서, 본 실시예의 확률적 스케줄링 방법에서는 확률적 분포의 누적 함수를 차함수로 표현하고 이를 확률적 계산량 모델로 이용한다.

[0016] 예를 들면, 단일 또는 멀티 코어 프로세서에 있어서, 일반적인 작업(task)은 입력되는 데이터의 양, 관련된 동작 모드, 명령어의 종류 등에 따라 도 2a에 도시한 바와 같은 확률적 분포(A_t)의 형태로 표현될 수 있다. 베스트 케이스(best-case)는 최소 실행 시간에 대응하고, 워스트 케이스(worst-case)는 최대 실행 시간에 대응한다.

[0017] 다음, 도 2a에 도시한 바와 같은 확률적 분포 함수를 도 2b에 도시한 바와 같은 누적 함수(cumulative probability)로 표현하면, 누적 함수의 차함수(tail probability)는 도 2c에 도시한 바와 같이 나타낼 수 있다.

[0018] 도 2b의 누적 함수는 아래의 수학적 식 1과 같이 표현되고, 도 2c의 차함수는 수학적 식 2와 같이 표현될 수 있다.

[0019] [수학적 식 1]

$$\int_0^t A_x dx$$

[0021] [수학적 식 2]

$$P_t = 1 - \int_0^t A_x dx$$

[0023] 여기에서, 차함수(P_t)는 특정 작업이 시간(t)에서 수행되고 있을 확률을 나타내는 것으로서 t의 값이 증가함에 따라 누적 함수가 항상 증가하므로 차함수는 항상 감소하는 특성을 있다. 본 발명의 실시예에서는 이러한 차함수를 확률적 계산량 모델로 이용한다.

[0024] 따라서, 본 실시예에서는 특정 작업에 대하여 도 2a에 도시한 바와 같은 확률적 분포(A_t)를 알 수 있도록 미리

데이터를 축적하여 두는 것도 바람직하다.

- [0025] 확률적 계산량 모델이 결정되면, 결정된 계산량 모델에 의해 특정 시간(t)에서 작업이 수행될 확률을 알 수 있으므로, 이에 기초하여 수행 대상 작업의 유틸리티를 최적화할 수 있도록 동작 모드를 결정(스케줄링)하게 된다.
- [0026] 여기에서, 유틸리티(Utility)는 QoS(Quality of Service)의 평균적인 기대값과 유사한 개념으로서, 높은 유틸리티를 갖는 동작 모드에서는 에너지 소모량이 크고 작업 수행 효율이 높으며 상대적으로 낮은 유틸리티를 갖는 동작 모드에서는 에너지 소모량이 적은 대신 작업 수행 효율은 낮은 것으로 정의할 수 있다. 작업 수행 효율이라 함은 예컨대 모터의 동작 속도, 데이터 전송 속도, 데이터 처리량, 중앙처리장치(CPU)의 동작 전압, 동작 클럭 속도 등과 같은 것으로서, 유틸리티가 높으면 동작 속도 등이 증가하지만 그에 상응하여 에너지 소비가 증가하게 된다.
- [0027] 본 발명의 실시예에서는 확률이 높은 계산 진행 전반부에서 낮은 실행 속도 또는 낮은 유틸리티를 적용하고, 확률이 낮은 계산 진행 후반부로 갈수록 높은 실행 속도 또는 높은 유틸리티를 적용할 수 있다. 이러한 구성에 대하여 아래에서 좀더 구체적으로 설명한다.
- [0028] 도 3a 및 도 3b는 본 발명의 일 실시예에 따른 확률적 스케줄링 방법에 채용가능한 병렬처리 기반의 확률적 계산량 모델을 설명하기 위한 도면들이다.
- [0029] 도 3a를 참조하면, 특정 작업에 대하여 두 개의 프로세서 코어들이 동일한 계산량으로 병렬처리하는 경우를 적용하면, 이론적으로 분배된 최대 계산량 또는 작업 종료시간은 절반으로 줄어든다. 도 3a에서 P^1 은 단일 코어에서의 계산량(# of Computation Cycles)이고, P^2 는 듀얼 코어에서의 병렬 처리 계산량이다.
- [0030] 그리고, 작업을 수행하는 코어의 개수에 따라 미래 계산량 확률(P_t)은 도 3b에 도시된 바와 같이 차 누적 함수(Tail Cumulative Probability)로 표현될 수 있다. 도 3b에서 P_t^1 은 단일 코어에서의 계산량 확률이고, P_t^2 는 듀얼 코어의 병렬 처리시의 계산량 확률이다.
- [0031] 도 4a 및 도 4b는 본 발명의 실시예에 따른 확률적 스케줄링 방법에 채용가능한 확률적 계산량 모델의 활용 방안을 설명하기 위한 도면들이다.
- [0032] 기존의 방법이 도 4a에 도시한 바와 같이 프로세스 코어의 실행 속도 예컨대 클럭 주파수(Clock Frequency)를 결정함에 있어서 테드라인에 수행 완료하도록 최대 실행 속도(worst-case)의 계산량 즉 최대 계산량을 활용한다고 할 때, 본 실시예에서는 도 4b에 도시한 바와 같이 미래 계산량 확률(P_t)에 근거하여 수행될 확률이 높은 계산 진행 전반부는 낮은 실행 속도로 수행하고, 수행될 확률이 낮은 계산 진행 후반부는 높은 실행 속도로 수행하도록 한다.
- [0033] 전술한 경우, 기존의 방법에서는 실제 계산량이 최대 실행 속도의 계산량보다 적기 때문에 낮은 실행 속도를 적용해서 프로세서의 에너지 소모율을 줄일 수 있는 기회를 가지지 못한다. 하지만, 본 실시예에서는 계산량의 확률(P_t)에 따라 전체 에너지 소모량의 확률적 기대값을 최소화하도록 계산량의 확률에 따라 실행 속도를 조절한다. 본 실시예에 의하면, 최대 실행 속도의 계산량을 기준으로 할 때 즉, 동일한 완료시간(Deadline)에 작업 수행을 종료한다고 할 때, 기존의 경우에 비해 에너지 소모량을 최소화할 수 있다.
- [0034] 전술한 본 발명의 일 실시예에 따른 확률적 스케줄링 방법을 수식으로 나타내면 다음과 같다.
- [0035] 본 실시예에 따른 확률적 스케줄링 방법에서는 주어진 작업에 대한 예상 에너지 소모량을 확률적으로 최소화하도록, 작업을 수행하는 코어의 개수와 계산 진행량에 따른 실행 속도 예컨대 동작 주파수를 결정한다.
- [0036] 이를 위해, 본 실시예의 확률적 스케줄링 방법에서는 몇 가지 기본적인 동작 원리를 적용한다. 첫째, 확률이 높은 계산 진행 후반부는 낮은 실행 속도를 적용하고, 확률이 낮은 계산 진행 후반부로 갈수록 높은 실행 속도를 적용한다. 둘째, 실행 속도 f_m 을 적용하다가 다음 실행 속도 $f_{(m+1)}$ 를 적용하는 시점을 $c[m]$ 이라 한다. 여기에서, $c[m]$ 은 $c[m+1]$ 과 같거나 작다. 셋째, 테드라인에 수행을 완료하면서 에너지 소모량의 확률적 기대값 즉 확률적 예상값을 최소화하도록 $c[m]$ 값들을 결정한다.
- [0037] 또한, 본 실시예에 따른 확률적 스케줄링 방법에서는 몇 가지 시스템 환경 변수를 적용한다. 첫째, N개의 코어들을 가진 프로세서를 가정하면, n개의 코어를 사용할 때 나머지 코어는 전원을 끈다. 여기에서, n은 1 이상 N

이하의 자연수이다. 둘째, n 개의 코어를 사용할 때, 계산량 확률 분포는 P^n , 최대(worst-case) 계산량은 W^n , 그리고 데드라인은 D 로 각각 표현된다. 셋째, M 개의 적용 가능한 실행 속도 예컨대 클럭 주파수는 각각 $f_1, f_2, \dots, f_M$ 으로 표현된다. 여기에서, M 은 2보다 큰 자연수이고, 각 실행 속도의 관계는 $f_1 < f_2 < \dots < f_M$ 이 된다. 넷째, f_m 의 실행속도에서 각 단계의 계산(a single cycle)을 수행하는데 필요한 에너지 소모량은 e_m 으로 표현된다. 여기에서, m 은 1 이상, M 이하의 자연수이며, 각 실행 속도에 대한 에너지 소모량의 관계는 $e_1 < e_2 < \dots < e_M$ 이 된다.

[0038] 작업을 수행하는 코어의 개수와 계산 진행량에 따른 실행 속도 예컨대 동작 주파수를 결정하기 위하여, 본 실시예의 확률적 스케줄링 방법은 고정된 n 의 값에 대하여 수학적 식 3에 따라 해법을 찾는다.

[0039] [수학적 식 3]

$$e_1 \cdot \int_0^{c[1]} P_t^n \cdot dt + e_2 \cdot \int_0^{c[2]} P_t^n \cdot dt + \dots + e_M \cdot \int_{c[M-1]}^{c[M]} P_t^n \cdot dt$$

[0040] 여기에서, 수학적 식 3은 아래의 수학적 식 4의 조건을 구비한다.

[0041] [수학적 식 4]

$$1/f_1 \cdot (c[1] - 0) + 1/f_2 \cdot (c[2] - c[1]) + \dots + 1/f_M \cdot (c[M] - c[M-1]) \leq D$$

[0042] 위의 수학적 식 3 및 수학적 식 4에 의해 N 개의 해법이 찾아지면, N 개의 해법 중에서 최소 에너지 소모량의 기대값을 가지는 최적의 코어 개수를 선택한다.

[0043] 작업을 수행하는 코어의 개수와 계산 진행량에 따른 실행 속도 예컨대 동작 주파수를 결정하는 과정을 좀더 구체적으로 설명하면 다음과 같다.

[0044] 도 5는 본 발명의 일 실시예에 따른 확률적 스케줄링 방법에서 비효율적 실행 속도를 제거하는 과정을 설명하기 위한 도면이다.

[0045] 도 5를 참조하면, 에너지 소모량에 비효율적인 실행 속도를 제거하기 위하여 본 실시예의 방법에서는 아래의 수학적 식 5와 같은 관계를 가지는 특정 실행 속도(f_j)를 제거한다.

[0046] [수학적 식 5]

$$\frac{(e_j - e_i)}{(1/f_i - 1/f_j)} > \frac{(e_k - e_j)}{(1/f_j - 1/f_k)}$$

[0047] 여기에서, f_j 는 $f_i < f_j < f_k$ 의 관계를 만족하며, i, j , 및 k 는 자연수이다.

[0048] 위의 과정에서 수학적 식 5의 관계를 가진 특정 실행 속도를 제거하고 남은 M 개의 실행 속도는 도 5에 도시한 바와 같이 아래의 수학적 식 6의 관계를 만족하게 된다.

[0049] [수학적 식 6]

$$\frac{[e_m - e_{(m-1)}]}{[1/f_{(m-1)} - 1/f_m]} \leq \frac{[e_{(m+1)} - e_m]}{[1/f_m - 1/f_{(m+1)}]}$$

[0050] 여기에서, 소정의 실행 속도 f_m 과 f_M 은 $f_1 < f_m < f_M$ 의 관계를 만족하며, m 및 M 은 자연수이다.

[0051] 다음, 남은 M 개의 실행속도들의 변환 시점을 아래의 수학적 식 7 또는 수학적 식 8에 기초하여 결정한다.

[0056] [수학식 7]

$$P_{c[x]} \cdot \frac{[e_{(x+1)} - e_x]}{[1/f_x - 1/f_{(x+1)}]} = P_{c[y]} \cdot \frac{[e_{(y+1)} - e_y]}{[1/f_y - 1/f_{(y+1)}]}$$

[0057]

[0058] [수학식 8]

$$P_{c[y]} = P_{c[x]} \cdot \frac{[e_{(x+1)} - e_x]}{[1/f_x - 1/f_{(x+1)}]} \cdot \frac{[1/f_y - 1/f_{(y+1)}]}{[e_{(y+1)} - e_y]}$$

[0059]

[0060] 수학식 7 및 수학식 8에서, $c[m]$ 은 변환 시점을 나타내고, m 은 1 이상, M 미만의 자연수이다.

[0061] 도 6a 및 도 6b는 본 발명의 일 실시예에 따른 확률적 스케줄링 방법에 채용가능한 실행속도의 변환시점을 찾는 방법을 설명하기 위한 도면들이다.

[0062] 도 6a에 도시한 바와 같이, 기준값 $c[1]$ 을 1부터 W 까지 차례로 고정하면서 위의 수학식 7 또는 수학식 8을 만족하는 $c[2]$, $c[3]$, ..., $c[M]$ 의 값들을 찾는다. 여기에서, $c[1]$ 을 고정하는 것은 수학식 7 또는 수학식 8에서 $P_{c[1]}$ 의 값을 고정하는 것에 대응한다.

[0063] 다음, 기준값 $c[1]=1$ 로 찾아진 $c[m]$ 값들을 사용한 수행 완료시간을 아래의 수학식 9를 이용하여 계산한다.

[0064] [수학식 9]

$$\sum_{1 \leq m \leq M} \frac{1}{f_2} \cdot (c[m-1] - c[m])$$

[0065]

[0066] 수학식 9의 계산 결과, 수행완료시간이 테드라인보다 크면 기준값을 $c[2]$ 로 교체한 후 $C[2]$ 를 1부터 W 까지 차례로 고정하면서 위의 수학식 7 또는 수학식 8을 만족하는 $c[3]$, $c[4]$, ..., $c[M]$ 의 값들을 찾는다.

[0067] 다시 말해서, 기준값 $c[b]=1$ 로 찾아진 $c[m]$ 값들을 사용한 수행완료시간이 테드라인보다 크면, 수행완료시간이 테드라인보다 작거나 같을 때까지 기준값을 $c[b+1]$ 로 변경하면서 위의 수학식 7 또는 수학식 8을 만족하는 $c[m]$ 값을 찾는다. 여기에서, $c[b+1]$ 로의 변경은 수학식 9에서 시작 실행 속도를 f_{b+1} 로 변경하는 것에 대응한다.

[0068] 전술한 기준값 $c[b]$ 는 도 6b에 도시한 바와 같이 이등분 검사(Bisection Search) 방식으로 선택될 수 있다. 사용가능한 $c[b]$ 값은 계산량(W)에 대응하는 폭의 중간 값을 선택하여 나머지 $c[b+1]$, ..., $c[M]$ 값들을 찾을 수 있다.

[0069] 또한, 찾아진 $c[m]$ 값들을 사용한 수행완료시간을 아래의 수학식 10을 이용하여 계산하고, 계산 결과를 테드라인(D) 값과 비교함으로써, 사용 가능한 $c[b]$ 값의 폭을 줄여갈 수 있다.

[0070] [수학식 10]

$$\sum_{b \leq m \leq M} \frac{1}{f_2} \cdot (c[m-1] - c[m])$$

[0071]

[0072] 본 실시예에서는 이등분 검사 방식을 사용하여 수행완료시간이 테드라인(D) 값과 같아지는 최적의 $c[b+1]$, ..., $c[M]$ 값들을 찾는다.

[0073] 다음, 최종적으로 찾아진 $c[b+1]$, ..., $c[M]$ 값들을 사용하여 아래의 수학식 11과 같이 예상 에너지 소모량 확률적 기대값을 계산한다.

[0074] [수학식 11]

$$n \cdot \sum_{b \leq m \leq M} e_m \cdot \int_{c[m-1]}^{c[m]} P_t^n \cdot dt$$

[0075] 여기서, $c[0]$ 은 0이다.

[0077] 다음으로, 각 n 에 대하여 위의 수학식 3을 이용하여 '에너지 소모량 확률적 기대값'을 계산한다. 여기서, n 은 1 이상, N 이하의 자연수이다. 즉, 본 단계에서는 하나의 프로세서 코어에서 프로세스 코어의 개수만큼 본 과정을 반복 수행하여 에너지 소모량 확률적 기대값을 계산한 후, 계산 결과를 토대로 최소 기대값을 가지는 n 을 선택한다.

[0078] 다음으로, 멀티 코어 프로세서에서는 n 의 값에 따라 P^n , 및 W^n 의 값도 변경되고, 사용하지 않는 $N-n$ 개의 코어의 전원은 오프(off)된다.

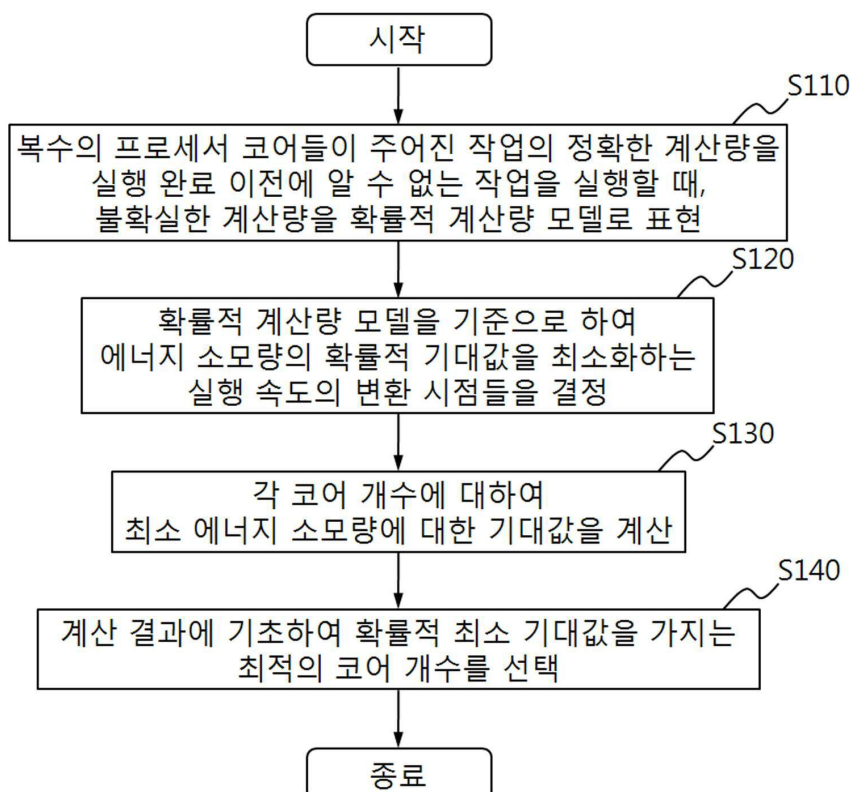
[0079] 한편, 멀티 코어 프로세서의 실행 속도 $f_b, \dots, f_M$ 은 n 개의 코어에 찾아진 $c[b], \dots, c[M]$ 값들에 따라 점진적으로 적용될 수 있다.

[0080] 전술한 실시예에 의하면, 확률적 계산량 모델을 기준으로 하여 에너지 소모량의 확률적 기대량을 최소화하는 코어 개수와 실행 속도를 스케줄링함으로써 멀티코어 프로세서가 저전력 작동을 할 수 있다.

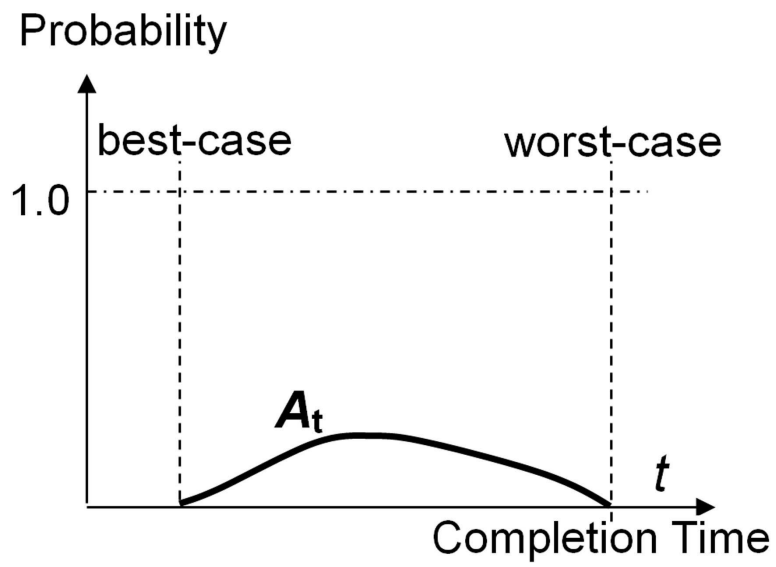
[0081] 이상에서, 바람직한 실시예들을 참조하여 본 발명을 설명하였으나, 본 발명은 상기의 실시예들에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능할 것이다. 따라서 본 발명은 첨부한 특허청구범위 및 도면 등의 전체적인 기재를 참조하여 해석되어야 할 것이며, 이의 균등 또는 등가적 변형 모두는 본 발명 사상의 범주에 속한다고 할 것이다.

도면

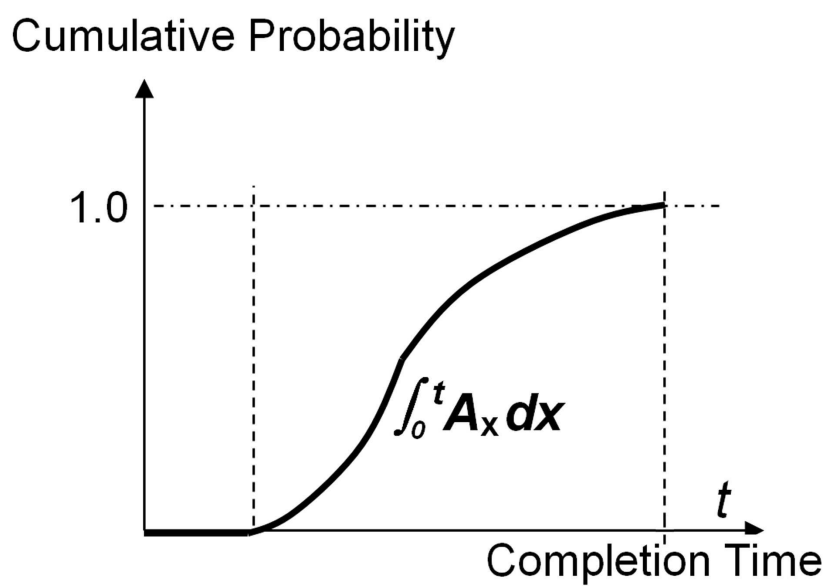
도면1



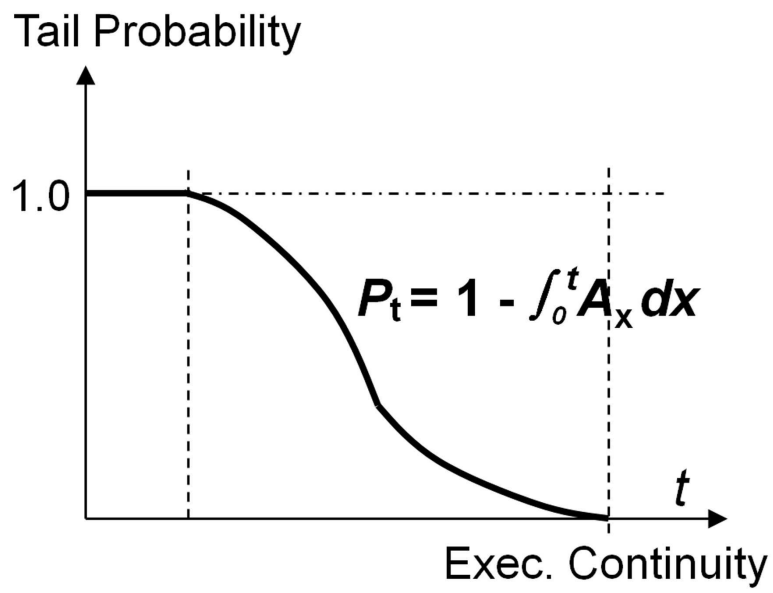
도면2a



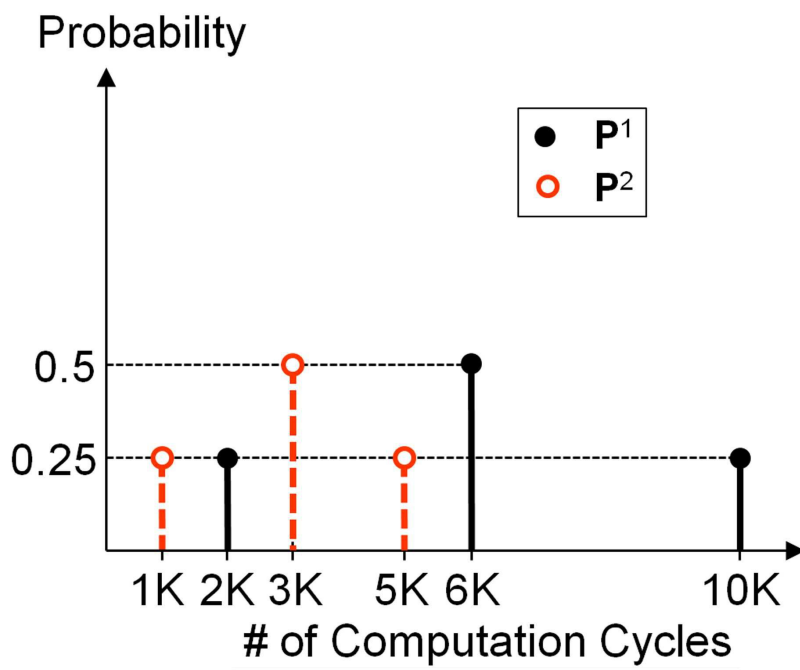
도면2b



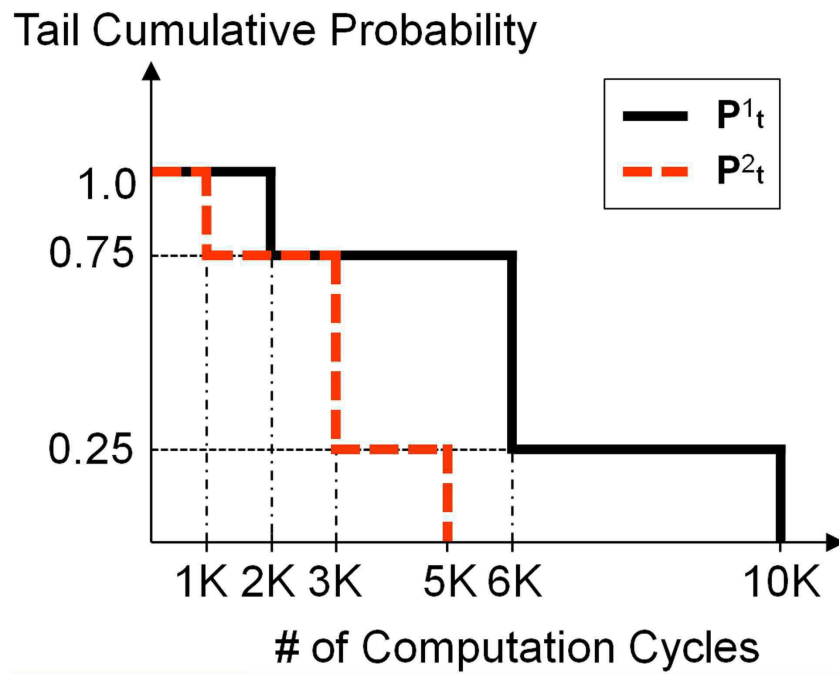
도면2c



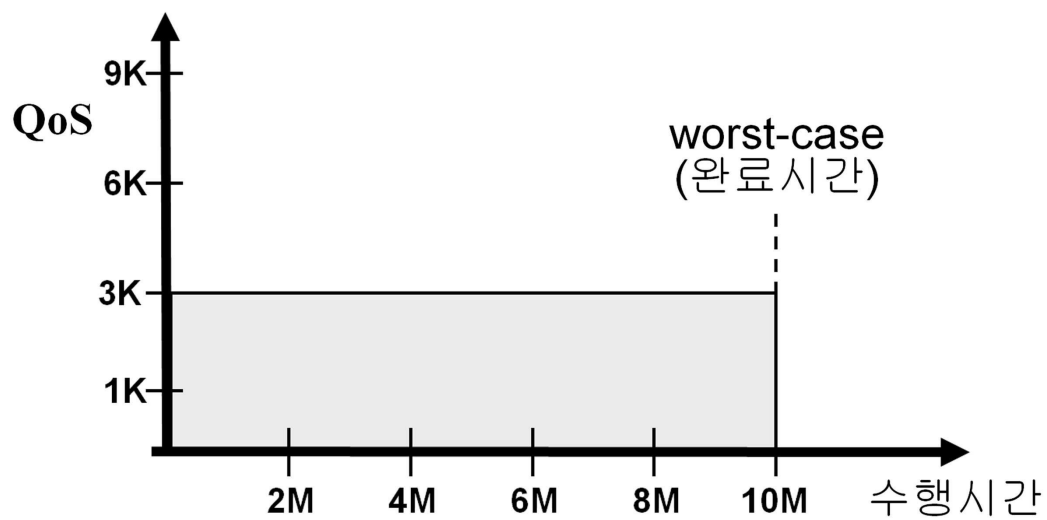
도면3a



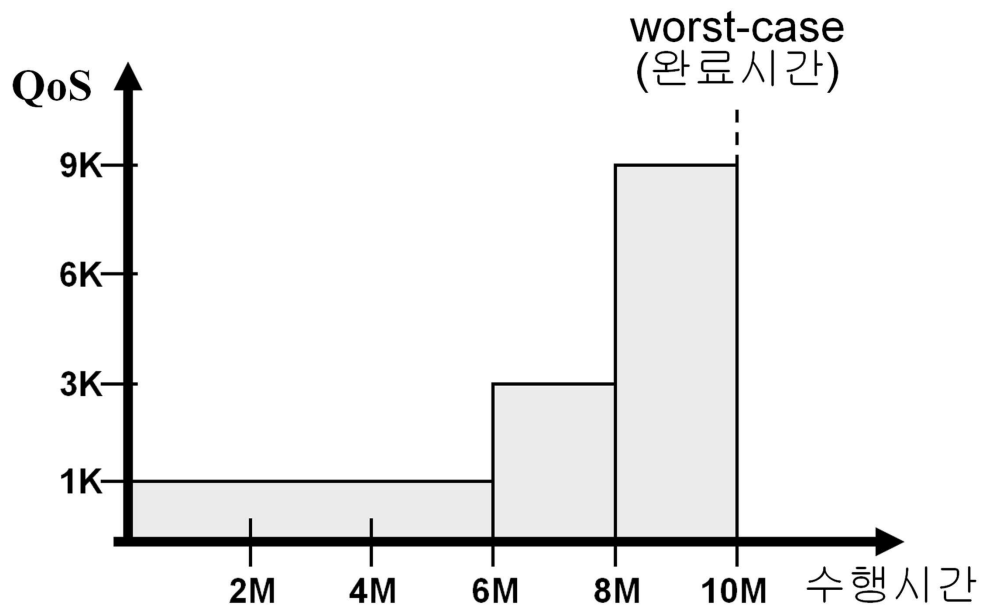
도면3b



도면4a

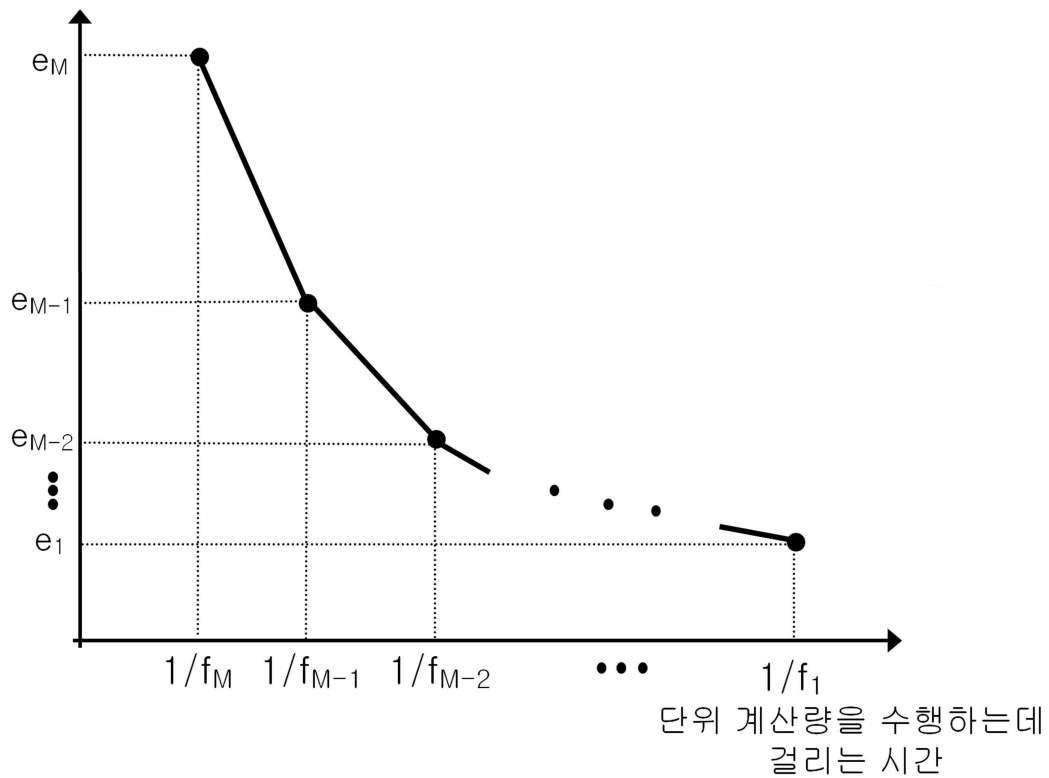


도면4b

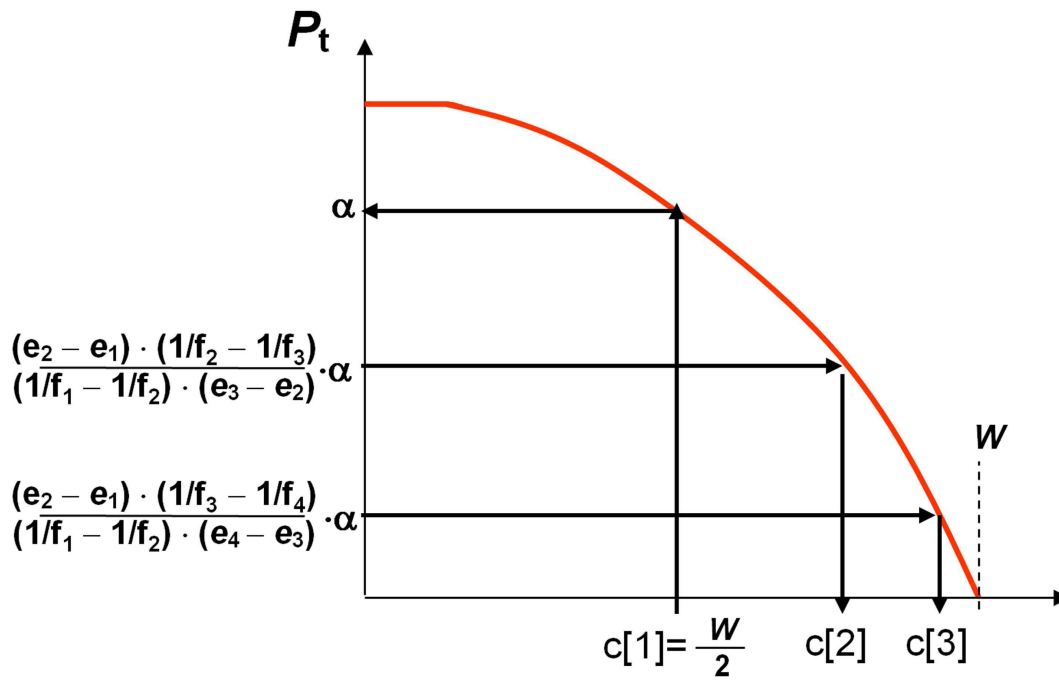


도면5

단위 계산량을 수행하는데
소요되는 에너지



도면6a



도면6b

